



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Ingeniería Técnica en Informática de Gestión
Proyecto Fin de Carrera



Análisis y Desarrollo de un Portal de Gestión Médica



Alumna: Yankoa López Cozas
Tutor: José Luís López Cuadrado

Junio, 2007

AGRADECIMIENTOS

En primer lugar, mis mayores agradecimientos van dedicados a mis padres, Pablo y Conchi, ya que gracias a ellos he llegado hasta donde estoy ahora, he conseguido poder realizar mis estudios además de recibir todo su apoyo y cariño durante todo este tiempo. Debo pedirles disculpas por el mal genio que he tenido durante esta época tan dura y por todo lo que han tenido que aguantarme, que es mucho, y por supuesto darles las gracias por el ánimo y la ayuda que me han proporcionado durante la consecución de este Proyecto Fin de Carrera.

En segundo lugar, tengo que darle las gracias a mi novio Carlos, porque él ha tenido que aguantar “carros y carretas” y soportar todos mis malos momentos. Sobre todo agradecerle que haya sabido comprenderme y apoyarme en todo momento, porque sin él todo esto se me habría hecho mucho más difícil de lo que ha sido. Solo decirle que esto ya toca su fin, y que por fin podremos recuperar todo el tiempo perdido.

En tercer lugar a mis compañeros de la empresa Ibermática, a los cuales debo agradecerles todo lo que me han enseñado y aportado, ya que gracias a ello surgió en mí la idea de hacer este Proyecto Fin de Carrera.

Por último, no quiero olvidarme del resto de mi familia y mi mejor amigo Josema, que también me han dado muchísimos ánimos y me han incentivado para continuar con mi carrera en los peores momentos.

A todos ellos muchísimas gracias.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS	7
ÍNDICE DE TABLAS	13
1.- INTRODUCCIÓN.....	14
1.1.- DESCRIPCIÓN.....	14
1.2.- ¿CÓMO SURGIÓ LA IDEA?.....	15
1.3.- OBJETIVOS.....	15
1.4.- ESTRUCTURA DE LA MEMORIA	16
2.- ESTADO DE LA CUESTIÓN	17
2.2.- HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS	19
2.2.1.- INTRODUCCIÓN.....	19
2.2.2.- INTRODUCCIÓN AL MARCO DE TRABAJO STRUTS	19
2.2.2.1.- INTRODUCCIÓN.....	19
2.2.2.1.1.- Modelo-Vista-Controlador y Struts.....	20
2.2.2.2.- CONTROLADOR DE STRUTS.....	21
2.2.2.2.1.- La Clase ActionServlet	22
2.2.2.2.2.- La Clase RequestProcessor	22
2.2.2.2.3.- La Clase Action	22
2.2.2.2.4.- La Clase ActionForward	23
2.2.2.3.- EL MODELO DE STRUTS	23
2.2.2.3.1.- Los Objetos de Negocio.....	23
2.2.2.3.2.- La Persistencia.....	24
2.2.2.3.3.- Patrones Bussiness Delegate y Dao	25
2.2.2.4.- LA VISTA DE STRUTS	25
2.2.2.4.1.- Uso de JavaBeans	26
2.2.2.4.2.- La Clase ActionForm	26
2.2.2.4.3.- Librerías de Etiquetas.....	27
2.2.2.5.- EL MARCO DE TRABAJO VALIDATOR.....	29
2.2.2.5.1.- El Archivo de Validation-Rules.xml.....	30
2.2.2.5.2.- El Archivo Validation.xml.....	31
2.2.2.6.- EL MARCO DE TRABAJO TILES	31
2.2.2.6.1.- La Librería de Etiquetas Tiles	32
2.2.2.6.2.- Uso de Definiciones	33
2.2.2.7.- CONFIGURAR APLICACIONES STRUTS	33
2.2.2.7.1.- Configurar el Archivo Web.xml para Struts.....	33
2.2.2.7.2.- Configurar el Archivo Struts-Config.xml	36
2.2.3.- INTRODUCCIÓN AL MARCO DE TRABAJO HIBERNATE.....	40
2.2.3.1.- INTRODUCCIÓN.....	40
2.2.3.2.- CONCEPTOS BÁSICOS DE HIBERNATE.....	41
2.2.3.3.- EL FICHERO DE CONFIGURACIÓN DE HIBERNATE.....	41
2.2.3.4.- EL FICHERO DE MAPEO DE HIBERNATE.....	43

2.2.3.4.1.- El Elemento <hibernate-mapping>	44
2.2.3.4.2.- El Elemento <class>	44
2.2.3.4.3.- Los Elementos <id> y <property>	44
2.2.3.4.4.- El Subelemento <generator>	44
2.2.3.4.5.- Mapear Asociaciones	45
2.2.3.5.- LAS SESIONES	48
2.2.3.6.- HIBERNATE QUERY LANGUAGE: HQL	49
2.2.4.- JAVA	50
2.2.4.1.- INTRODUCCIÓN.....	50
2.2.4.2.- JAVA 2 PLATFORM, STANDARD EDITION (J2SE)	51
2.2.4.3.- JAVA 2 PLATFORM, ENTERPRISE EDITION (J2EE)	51
2.2.4.4.- JAVA 2 PLATFORM, MICRO EDITION (J2ME).....	52
2.2.4.5.- COMENTARIOS	52
2.2.5.- MYSQL	52
2.2.5.1.- INTRODUCCIÓN.....	52
2.2.5.2.- USO DE MYSQL	53
2.2.5.3.- TIPOS DE DATOS	53
2.2.5.4.- COMENTARIOS	54
2.2.6.- APACHE TOMCAT	55
2.2.6.1.- INTRODUCCIÓN.....	55
2.2.6.2.- FICHEROS DE CONFIGURACIÓN	55
2.2.6.3.- COMENTARIOS	56
2.2.7.- JAVA SERVER PAGES (JSP)	56
2.2.8.- JAVA SERVER PAGES STANDARD TAG LIBRARY (JSTL)	57
2.2.9.- JAVASCRIPT	59
2.2.10.- ECLIPSE WEB TOOLS PLATFORM (WTP)	60
2.2.10.1.- INTRODUCCIÓN.....	60
2.2.10.2.- CARACTERÍSTICAS	60
2.2.10.3.- WTP.....	61
2.2.10.4.- COMENTARIOS	63
2.2.11.- SQLYOG	63
2.2.12.- OTRAS HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS	63
 3.- GESTIÓN DEL PROYECTO.....	 64
3.1.- CICLO DE VIDA	64
3.2.- ORGANIZACIÓN DEL PROYECTO	65
3.2.1.- WORK BREAKDOWN STRUCTURE (WBS)	65
3.2.1.1.- DESCRIPCIÓN DE LAS TAREAS Y ACTIVIDADES DEL ESTÁNDAR	65
IEEE1074.....	65
3.2.1.2.- ESTRUCTURACIÓN DE TAREAS	71
3.2.1.3.- ACTIVIDADES Y TAREAS QUE NO SE REALIZAN	76
3.2.2.- RESOURCE BREAKDOWN STRUCTURE (RBS)	77
3.2.3.- PRODUCTOS A OBTENER EN CADA TAREA (PBS)	80
3.2.4.- PLANIFICACIÓN	87
3.2.5.- ASIGNACIÓN DE RECURSOS	90
3.2.6.- ESTIMACIÓN DE COSTES	95

4.- ANÁLISIS Y DISEÑO DEL PROBLEMA	97
4.1.- DIAGRAMA DE CASOS DE USO	97
4.1.1.- DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO	100
4.2.- FUNCIONALIDAD	109
4.3.- DIAGRAMAS DE ACTIVIDAD	115
4.4.- DIAGRAMA DE CLASES	134
4.4.1.- COMENTARIOS	136
4.5.- DIAGRAMAS DE SECUENCIA	136
4.6.- DIAGRAMA E/R	149
4.6.1.- SUPUESTOS SEMÁNTICOS	150
4.7.- MODELO RELACIONAL	151
 5.- PROTOTIPO: INSTALACIÓN Y MANUAL DE USUARIO	 155
5.1. INSTALACIÓN	155
5.1.1.- INSTALACIÓN DE JDK	155
5.1.2.- INSTALACIÓN DE MYSQL 5.0	159
5.1.3.- DESCARGA DEL DRIVER PARA MYSQL 5.0	166
5.1.4.- INSTALACIÓN DE SQLYOG 5.21	167
5.1.4.1.- IMPORTACIÓN DE LA BASE DE DATOS DEL CENTRO DE SALUD	172
5.1.5.- INSTALACIÓN DE APACHE TOMCAT 5.5	173
5.1.6.- INSTALACIÓN DE ECLIPSE WTP 1.5.2	178
5.1.6.1.- CONFIGURACIÓN	179
5.2.- MANUAL DE USUARIO	190
5.2.1.- ESTRUCTURA DE LOS INTERFACES	190
5.2.2.- ¿CÓMO EMPEZAR?	191
5.2.3.- LOGEARSE COMO ADMINISTRADOR Ó MÉDICO	192
5.2.4.- OPCIONES DEL ADMINISTRADOR	195
5.2.4.1.- DAR DE ALTA UN NUEVO ADMINISTRADOR	195
5.2.4.2.- MODIFICAR PERFIL	197
5.2.4.3.- DAR DE ALTA UN NUEVO MÉDICO	198
5.2.4.4.- CONSULTAR MÉDICO	200
5.2.4.5.- DAR DE ALTA UN NUEVO PACIENTE	203
5.2.4.6.- CONSULTAR PACIENTE	209
5.2.4.7.- CITAR PACIENTE	212
5.2.4.8.- DAR DE ALTA UN NUEVO MEDICAMENTO	215
5.2.4.9.- CONSULTAR MEDICAMENTO	216
5.2.4.10.- DAR DE ALTA UNA NUEVA VACUNA	219
5.2.4.11.- MODIFICAR VACUNA	221
5.2.4.12.- DAR DE ALTA UNA NUEVA NOTICIA	223
5.2.4.13.- MODIFICAR NOTICIA	225
5.2.4.14.- DAR DE ALTA UN NUEVO ASUNTO DEL FORO	226
5.2.4.15.- BORRAR UN ASUNTO Ó PREGUNTA DEL FORO	228
5.2.4.16.- CAMBIAR MAPA DEL CENTRO	231
5.2.5.- OPCIONES DEL MÉDICO	233
5.2.5.1.- CONSULTAR PACIENTE	233
5.2.5.2.- GENERAR DIAGNÓSTICO	234

5.2.5.3.- CONSULTAR HISTORIAL	241
5.2.5.4.- RECETAR MEDICAMENTO	244
5.2.5.5.- CITAR PACIENTE	245
5.2.5.6.- CONSULTAR MEDICAMENTO	246
5.2.5.7.- CONSULTAR AGENDA	248
5.2.5.8.- CONSULTAR FORO Y/O RESPONDER PREGUNTA	250
5.2.6.- CERRAR SESIÓN	255
5.2.7.- OPCIONES DE CUALQUIER USUARIO NO LOGADO	255
5.2.7.1.- VISUALIZAR DIRECTORIO DE MÉDICOS	255
5.2.7.2.- VISUALIZAR MAPA DEL CENTRO	257
5.2.7.3.- CALENDARIO DE VACUNACIONES	258
5.2.7.4.- CONSULTAR FORO Y/O FORMULAR PREGUNTA	259
 6.- ACCESIBILIDAD	 264
6.1.- INTRODUCCIÓN	264
6.2.- PAUTAS DE ACCESIBILIDAD	265
6.2.1.- PAUTA 1	265
6.2.2.- PAUTA 2	265
6.2.3.- PAUTA 3	267
6.2.4.- PAUTA 4	267
6.2.5.- PAUTA 5	267
6.2.6.- PAUTA 6	268
6.2.7.- PAUTA 7	268
6.2.8.- PAUTA 8	268
6.2.9.- PAUTA 9	268
6.2.10.- PAUTA 10	268
6.2.11.- PAUTA 11	269
6.2.12.- PAUTA 12	269
6.2.13.- PAUTA 13	269
6.2.14.- PAUTA 14	269
6.3.- ANÁLISIS DE ACCESIBILIDAD	269
6.3.1.- VERIFICACIÓN DE HOJAS DE ESTILO EN CASCADA	270
6.3.2.- ANÁLISIS DE LAS PÁGINAS DE ACCESO PÚBLICO	274
6.3.2.1.- ANÁLISIS DE LA PÁGINA DE BIENVENIDA	276
6.3.2.2.- ANÁLISIS DE LAS PÁGINAS DE VISUALIZACIÓN DEL DIRECTORIO DE MÉDICOS	279
6.3.2.3.- ANÁLISIS DE LA PÁGINA VISUALIZAR MAPA DEL CENTRO	283
6.3.2.6.- ANÁLISIS DE LAS PÁGINAS DEL FORO	291
6.3.2.7.- ANÁLISIS DE LA PÁGINA DE LOGIN	299
6.3.3.- ANÁLISIS DE LAS PÁGINAS DE ACCESO PRIVADO	301
6.3.3.1.- ANÁLISIS DE UNA PÁGINA DE ALTA	302
 7.- CONCLUSIONES Y LÍNEAS FUTURAS	 308
7.1.- CONCLUSIONES	308
7.2.- FUTURAS LÍNEAS DE TRABAJO	308

8.- BIBLIOGRAFÍA.....	310
------------------------------	------------

ANEXO	312
--------------------	------------

ÍNDICE DE FIGURAS

- Figura 1: Modelo1 -	20
- Figura 2: Modelo 2 (MVC) -	20
- Figura 3: Componentes del controlador de Struts -	21
- Figura 4: Motor de persistencia -	24
- Figura 5: Ejemplo de tiles de una aplicación Web -	32
- Figura 6: Arquitectura de Hibernate -	41
- Figura 7: Relación n-m -	46
- Figura 8: Relación 1-n -	48
- Figura 9: Ediciones Java -	51
- Figura 10: Alcance de los subproyectos de WTP -	62
- Figura 11: Alcance WBS -	78
- Figura 12: Diagrama de Gantt 1 -	87
- Figura 13: Diagrama de Gantt 2 -	88
- Figura 14: Diagrama de Gantt 3 -	88
- Figura 15: Diagrama de Gantt 4 -	88
- Figura 16: Diagrama de Gantt 5 -	89
- Figura 17: Diagrama de Gantt 6 -	89
- Figura 18: Diagrama de Gantt 7 -	89
- Figura 19: Diagrama de casos de uso 1 -	98
- Figura 20: Diagrama de casos de uso 2 -	99
- Figura 21: Diagrama actividad identificarse -	115
- Figura 22: Diagrama actividad modificar datos paciente -	115
- Figura 23: Diagrama actividad consultar datos personales paciente -	116
- Figura 24: Diagrama actividad consultar agenda -	116
- Figura 25: Diagrama actividad generar diagnóstico -	117
- Figura 26: Diagrama actividad generar diagnóstico -	117
- Figura 27: Diagrama actividad realizar seguimiento de Sintrom -	118
- Figura 28: Diagrama actividad recetar medicamento -	119
- Figura 29: Diagrama actividad consultar medicamento -	120
- Figura 30: Diagrama actividad consultar foro -	120
- Figura 31: Diagrama actividad responder al foro -	121
- Figura 32: Diagrama actividad citar paciente -	121
- Figura 33: Diagrama actividad consultar historial -	122
- Figura 34: Diagrama actividad consultar calendario de vacunaciones -	122
- Figura 35: Diagrama actividad preguntar en el foro -	123
- Figura 36: Diagrama actividad visualizar directorio -	123
- Figura 37: Diagrama actividad consultar noticia -	123
- Figura 38: Diagrama actividad visualizar mapa del centro -	124
- Figura 39: Diagrama actividad dar alta administrador -	124
- Figura 40: Diagrama actividad dar alta administrador -	124
- Figura 41: Diagrama actividad modificar perfil -	125
- Figura 42: Diagrama actividad dar alta médico -	125
- Figura 43: Diagrama actividad dar alta médico -	126
- Figura 44: Diagrama actividad consultar médico -	126
- Figura 45: Diagrama actividad dar alta paciente -	127
- Figura 46: Diagrama actividad dar alta medicamento -	127
- Figura 47: Diagrama actividad dar alta paciente -	128
- Figura 48: Diagrama actividad dar alta medicamento -	128

- Figura 49: Diagrama actividad generar cita -	129
- Figura 50: Diagrama actividad dar alta vacuna -	129
- Figura 51: Diagrama actividad cambiar calendario vacunaciones -	130
- Figura 52: Diagrama actividad cambiar mapa del centro -	130
- Figura 53: Diagrama actividad dar alta noticia -	131
- Figura 54: Diagrama actividad dar alta noticia -	131
- Figura 55: Diagrama actividad borrar noticia -	131
- Figura 56: Diagrama actividad crear asunto -	132
- Figura 57: Diagrama actividad crear asunto -	132
- Figura 58: Diagrama actividad borrar asunto -	133
- Figura 59: Diagrama actividad borrar pregunta	133
- Figura 60: Diagrama de clases I -	134
- Figura 61: Diagrama de clases II -	135
- Figura 62: Diagrama de secuencia identificarse -	137
- Figura 63: Diagrama de secuencia consultar datos personales y modificar datos paciente -	137
- Figura 64: Diagrama de secuencia consultar agenda -	138
- Figura 65: Diagrama de secuencia recetar medicamento -	138
- Figura 66: Diagrama de secuencia citar paciente -	139
- Figura 67: Diagrama de secuencia generar diagnóstico y seguimiento de Sintrom -	140
- Figura 68: Diagrama de secuencia consultar y modificar medicamento -	141
- Figura 69: Diagrama de secuencia consultar historial -	141
- Figura 70: Diagrama de secuencia consultar foro -	142
- Figura 71: Diagrama de secuencia responder al foro -	142
- Figura 72: Diagrama de secuencia preguntar en el foro -	142
- Figura 73: Diagrama de secuencia visualizar directorio -	143
- Figura 74: Diagrama de secuencia consultar noticia -	143
- Figura 75: Diagrama de secuencia dar de alta administrador -	144
- Figura 76: Diagrama de secuencia modificar perfil -	144
- Figura 77: Diagrama de secuencia dar alta médico -	144
- Figura 78: Diagrama de secuencia consultar datos personales y modificar médico -	145
- Figura 79: Diagrama de secuencia dar de alta paciente -	145
- Figura 80: Diagrama de secuencia generar cita -	145
- Figura 81: Diagrama de secuencia dar de alta medicamento -	146
- Figura 82: Diagrama de dar alta vacuna -	146
- Figura 83: Diagrama de secuencia cambiar calendario de vacunaciones -	146
- Figura 84: Diagrama de secuencia dar alta noticia -	147
- Figura 85: Diagramas de secuencia cambiar mapa del centro -	147
- Figura 86: Diagrama de secuencia crear asunto -	148
- Figura 87: Diagrama de secuencia borrar asunto -	148
- Figura 88: Diagrama de secuencia borrar pregunta -	148
- Figura 89: Diagrama E/R -	149
- Figura 90: Instalación JDK 1 -	156
- Figura 91: Instalación JDK 2 -	156
- Figura 92: Instalación JDK 3 -	157
- Figura 93: Instalación JDK 4 -	157
- Figura 94: Instalación JDK 5 -	158
- Figura 95: Descargar MySQL -	159

- Figura 96: Instalación de MySQL 1 -	160
- Figura 97: Instalación de MySQL 2 -	160
- Figura 98: Instalación de MySQL 3 -	161
- Figura 99: Instalación de MySQL 4 -	162
- Figura 100: Instalación de MySQL 5 -	162
- Figura 101: Instalación de MySQL 6 -	163
- Figura 102: Instalación de MySQL 7 -	163
- Figura 103: Instalación de MySQL 8 -	164
- Figura 104: Instalación de MySQL 9 -	165
- Figura 105: Instalación de MySQL 10 -	165
- Figura 106: Instalación de MySQL 11 -	166
- Figura 107: Instalación de SQLYog 1 -	167
- Figura 108: Instalación de SQLYog 2 -	168
- Figura 109: Instalación de SQLYog 3 -	168
- Figura 110: Instalación de SQLYog 4 -	169
- Figura 111: Instalación de SQLYog 5 -	170
- Figura 112: Instalación de SQLYog 6 -	170
- Figura 113: Instalación de SQLYog 7 -	171
- Figura 114: Instalación de SQLYog 8 -	172
- Figura 115: Instalación de SQLYog 9 -	172
- Figura 116: Importar datos BD -	173
- Figura 117: Instalación de Tomcat 1 -	174
- Figura 118: Instalación de Tomcat 2 -	174
- Figura 119: Instalación de Tomcat 3 -	175
- Figura 120: Instalación de Tomcat 5 -	176
- Figura 122: Instalación de Tomcat 7 -	177
- Figura 121: Instalación de Tomcat 6 -	177
- Figura 123: Instalación de Tomcat 8 -	178
- Figura 124: Configuración de Eclipse WTP 1 -	179
- Figura 125: Borrado de la carpeta Servers -	180
- Figura 126: Configuración de Eclipse WTP 2 -	181
- Figura 127: Configuración de Eclipse WTP 3 -	182
- Figura 128: Configuración de Eclipse WTP 4 -	183
- Figura 129: Configuración de Eclipse WTP 5 -	183
- Figura 130: Configuración de Eclipse WTP 6 -	184
- Figura 131: Configuración de Eclipse WTP 7 -	185
- Figura 132: Configuración de Eclipse WTP 8 -	185
- Figura 133: Configuración de Eclipse WTP 9 -	186
- Figura 134: Configuración de Eclipse WTP 10 -	186
- Figura 135: Configuración de Eclipse WTP 11 -	187
- Figura 136: Configuración de Eclipse WTP 12 -	188
- Figura 137: Configuración de Eclipse WTP 13 -	189
- Figura 138: Configuración de Eclipse WTP 14 -	189
- Figura 139: Configuración Interfaz del administrador-	190
- Figura 140: Interfaz del médico-	190
- Figura 141: Interfaz del usuario-	191
- Figura 142: Bienvenida -	192
- Figura 143: Acceso profesional -	193
- Figura 144: Acceso incorrecto -	194
- Figura 145: Bienvenida administrador -	194

- Figura 146: Bienvenida médico -	195
- Figura 147: Alta administrador-.....	196
- Figura 148: Administrador guardado con éxito-	197
- Figura 149: Modificar perfil -	198
- Figura 150: Alta médico -.....	199
- Figura 151: Médico guardado con éxito -	200
- Figura 152: Consultar médico -	201
- Figura 153: Consultar médico introduciendo algún criterio -	202
- Figura 154: Mostrar todos los médicos -.....	202
- Figura 155: Búsqueda de médico sin coincidencias -	203
- Figura 156: Alta paciente -	204
- Figura 157: Paciente guardado con éxito -.....	205
- Figura 158: Añadir alergia -.....	206
- Figura 159: Datos paciente con alergia -.....	207
- Figura 160: Asignar médico -.....	208
- Figura 161: Datos paciente con alergia y médico asignado -	209
- Figura 162: Consultar paciente -	210
- Figura 163: Consultar paciente introduciendo algún criterio -	211
- Figura 164: Mostrar todos los pacientes -.....	211
- Figura 165: Búsqueda de paciente sin coincidencia -.....	212
- Figura 166: Administrador cita a paciente -.....	213
- Figura 167: Error al citar al paciente -	214
- Figura 168: Volante de cita -	214
- Figura 169: Alta medicamento -	215
- Figura 170: Medicamento guardado con éxito -.....	216
- Figura 171: Consultar medicamento -.....	217
- Figura 172: Consultar medicamento introduciendo algún criterio -	218
- Figura 173: Mostrar todos los medicamentos -	218
- Figura 174: Búsqueda de medicamento sin coincidencias -	219
- Figura 175: Alta vacuna -.....	220
- Figura 176: Vacuna guardada con éxito -.....	221
- Figura 177: Consultar vacuna -	222
- Figura 178: Vacuna seleccionada -	223
- Figura 179: Alta noticia -.....	224
- Figura 180: Noticia guardada con éxito -.....	225
- Figura 181: Listado de noticias -.....	226
- Figura 182: Alta asunto -.....	227
- Figura 183: Asunto guardado con éxito -.....	228
- Figura 184: Listado de asuntos -.....	229
- Figura 185: Listado de preguntas -.....	230
- Figura 186: Listado de respuestas -.....	231
- Figura 187: Cambiar mapa -.....	232
- Figura 188: Visualizar mapa -.....	233
- Figura 189: Generar diagnóstico -.....	234
- Figura 190: Datos diagnóstico -.....	235
- Figura 191: Diagnóstico guardado con éxito -	236
- Figura 192: Receta -.....	237
- Figura 193: Médico cita a paciente -.....	238
- Figura 194: Seguimiento Sintrom -.....	239
- Figura 195: Seguimiento Sintrom guardado con éxito -.....	240

- Figura 196: DTS < 0,5 -	240
- Figura 197: Consultar historial -	241
- Figura 198: Consultar historial introduciendo algún criterio -	242
- Figura 199: Mostrar todos los historiales -	242
- Figura 200: Listado de diagnósticos -	243
- Figura 201: Consultar diagnóstico -	244
- Figura 202: Recetar medicamento -	245
- Figura 203: Médico cita a paciente -	246
- Figura 204: Consultar medicamento -	247
- Figura 205: Visualizar medicamento -	248
- Figura 206: Consultar agenda -	249
- Figura 207: Visualizar citas -	250
- Figura 208: Listado de asuntos -	251
- Figura 209: Listado de preguntas -	252
- Figura 210: No existe ninguna respuesta -	253
- Figura 211: Responder -	254
- Figura 212: Respuesta guardada con éxito -	255
- Figura 213: Listado de médicos del centro -	256
- Figura 214: Visualizar datos del médico -	257
- Figura 215: Visualizar mapa -	258
- Figura 216: Calendario de vacunaciones -	259
- Figura 218: Preguntar -	261
- Figura 217: Listado de preguntas -	260
- Figura 219: Pregunta guardada con éxito -	262
- Figura 220: Listado de respuestas -	263
- Figura 221: Analizador de contrastes 1 -	266
- Figura 222: Analizador de contrastes 2 -	266
- Figura 223: Validación de CSS 1 -	270
- Figura 224: Validación de CSS 2 -	271
- Figura 225: Validación de CSS 3 -	272
- Figura 226: Resultado validación de estilos.css -	273
- Figura 227: Resultado validación de calendario.css -	274
- Figura 228: Test de accesibilidad -	275
- Figura 229: Bienvenida -	276
- Figura 230: Resultado accesibilidad Pág. de bienvenida -	277
- Figura 231: Resultado accesibilidad Pág. de bienvenida prioridad 1 -	278
- Figura 232: Resultado accesibilidad Pág. de bienvenida prioridad 2 -	278
- Figura 233: Resultado accesibilidad Pág. de bienvenida prioridad 3 -	279
- Figura 234: Visualizar directorio -	279
- Figura 235: Resultado accesibilidad Pág. visualizar directorio -	280
- Figura 236: Visualizar datos médicos -	281
- Figura 237: Resultado accesibilidad Pág. visualizar datos médicos -	282
- Figura 238: Visualizar mapa del centro -	283
- Figura 239: Resultado accesibilidad Pág. visualizar mapa del centro -	284
- Figura 240: Calendario de vacunaciones -	285
- Figura 241: Resultado accesibilidad Pág. calendario de vacunaciones -	286
- Figura 242: Consultar noticias -	287
- Figura 243: Resultado accesibilidad Pág. consultar noticias -	288
- Figura 244: Visualizar noticia -	289
- Figura 245: Resultado accesibilidad Pág. visualizar noticia -	290

- Figura 246: Resultado accesibilidad Pág. visualizar noticia prioridad 2 -	290
- Figura 247: Listado de asuntos -	291
- Figura 248: Resultado accesibilidad Pág. listado de asuntos -	292
- Figura 249: Visualizar preguntas -	293
- Figura 250: Resultado accesibilidad Pág. visualizar preguntas -	294
- Figura 251: Resultado accesibilidad Pág. visualizar preguntas prioridad 2 -	294
- Figura 252: Preguntar -	295
- Figura 253: Resultado accesibilidad Pág. preguntar -	296
- Figura 254: Resultado accesibilidad Pág. preguntar prioridad 2 -	296
- Figura 255: Visualizar respuesta -	297
- Figura 256: Resultado accesibilidad Pág. visualizar respuestas -	298
- Figura 257: Resultado accesibilidad Pág. visualizar respuestas prioridad 2 - ...	298
- Figura 258: Login -	299
- Figura 259: Resultado accesibilidad Pág. de login -	300
- Figura 260: Resultado accesibilidad Pág. de login prioridad 2 -	300
- Figura 261: Alta administrador-	302
- Figura 262: Resultado accesibilidad Pág. alta administrador -	303
- Figura 263: Resultado accesibilidad Pág. alta administrador prioridad 2 -	303
- Figura 264: Recuperar vacuna -	304
- Figura 265: Resultado accesibilidad Pág. recuperar vacuna -	305
- Figura 266: Resultado accesibilidad Pág. recuperar vacuna prioridad 2 -	305
- Figura 267: Consultar medicamento -	306
- Figura 268: Resultado accesibilidad Pág. consultar medicamento -	307
- Figura 269: Jerarquía usuario -	312
- Figura 270: Clase paciente -	312
- Figura 271: Clase alergia -	313
- Figura 272: Clase diagnóstico -	313
- Figura 273: Clase seguimiento de Sintrom -	313
- Figura 274: Clase medicamento -	314
- Figura 275: Clase noticia -	314
- Figura 276: Clase vacuna -	314
- Figura 277: Clase asunto -	314
- Figura 278: Clase pregunta -	315

ÍNDICE DE TABLAS

- Tabla 1: Cronología de Internet en España -	18
- Tabla 2: Algunas etiquetas de la librería HTML -	28
- Tabla 3: Algunas etiquetas de la librería Bean -	28
- Tabla 4: Algunas etiquetas de la librería Logic -	29
- Tabla 5: Reglas de validación en la clase GenericValidator -	31
- Tabla 6: Etiquetas de la librería de etiquetas Tiles -	32
- Tabla 7: Elementos hijos del elemento servlet -	34
- Tabla 8: Elementos hijos del elemento taglib -	34
- Tabla 9: Atributos para el elemento data-source -	36
- Tabla 11: Atributos del elemento exception -	37
- Tabla 10: Atributos del elemento form-bean -	37
- Tabla 12: Atributos del elemento forward -	38
- Tabla 13: Algunos atributos del elemento action -	39
- Tabla 14: Atributos del elemento message-resources -	39
- Tabla 15: Dialectos de Hibernate -	43
- Tabla 16: MySQL datos numéricos -	53
- Tabla 17: MySQL datos tipo fecha -	54
- Tabla 18: MySQL datos tipo cadena -	54
- Tabla 19: Elementos del fichero Server.xml -	56
- Tabla 20: Etiquetas Core -	58
- Tabla 21: Etiquetas Fmt -	58
- Tabla 22: Actividades y tareas -	76
- Tabla 23: Asignación de recursos -	94
- Tabla 24: Asignación de presupuestos -	95
- Tabla 25: Estimación del coste -	96



1.- INTRODUCCIÓN

1.1.- DESCRIPCIÓN

Este proyecto está basado en el análisis, diseño y desarrollo, de una aplicación Web destinada a un Centro de Salud de atención primaria español, a través de la cual se permita el desarrollo y la gestión de dicho centro, gracias a la cooperación de diversos elementos.

La aplicación esta diseñada para su uso por los médicos y administradores del centro, y para cualquier usuario que acceda a ella, a través de Internet. De esta forma diferenciamos tres roles:

En primer lugar tendremos al administrador de la aplicación, pudiendo existir por supuesto más de uno. El administrador tendrá encomendadas diversas tareas mediante las cuales mantendrá la aplicación bajo un correcto funcionamiento y una actualización continua, así, este se encargará entre otras cosas, de dar de alta nuevos miembros en el centro de salud, administrar los medicamentos, moderar el foro de debate, administrar las noticias...

Por otro lado, nos encontraremos con el perfil de médico, pudiendo existir diversos médicos, eso sí, todos ellos siendo médicos de familia cuyas funciones por lo tanto son las mismas. Principalmente los médicos se encargarán de la administración de los pacientes y sus historiales, así podrán entre otros aspectos, consultar los datos de los pacientes, consultar el historial de un paciente, recetar medicamentos, citar a los pacientes...

Finalmente la aplicación podrá ser accedida por cualquier internauta que teclee la dirección de esta aplicación a través de su navegador Web. Los internautas o usuarios como los llamaremos a lo largo de este documento, únicamente podrán realizar diversas labores de consulta, por ejemplo, consultar el mapa del centro, el directorio de médicos, las noticias... Además tendrán la posibilidad de realizar preguntas en el foro, las cuales serán respondidas por los médicos del Centro de Salud.

Para acceder a dicha aplicación, es necesario introducir la dirección Web en el navegador. Una vez dentro de ella, existe la posibilidad de identificarse, lo cual solo es válido para médicos y administradores. Con esto, lo que se pretende indicar, es que, cualquiera puede actuar como usuario, es decir, los médicos y administradores pueden realizar cualquiera de las acciones de un usuario normal de la Web además de las suyas propias.

La aplicación se encuentra implementada a través de Java (J2EE), Struts, Hibernate, MySql y diversos patrones de diseño, mediante los cuales se mejora la estructura de la aplicación y se consigue que sea más amigable.



1.2.- ¿CÓMO SURGIÓ LA IDEA?

Tras diversas reuniones con profesores de la Universidad Carlos III de Madrid, descubrí que la mayoría de Proyectos Fin de Carrera que se ofertaban no me satisfacían.

Una vez me reuní con el tutor de este proyecto, José Luís López Cuadrado, me sugirió la idea de realizar un pequeño programa para poder realizar un control del acenocumarol (Sintrom). Esto me llamó la atención, pero me parecía escaso, así que le propuse al tutor realizar una aplicación Web para un Centro de Salud, en la que además del análisis de Sintrom, se pudiera gestionar dicho centro, permitiendo la interacción de todos los elementos que lo conforman (médicos, administradores y pacientes).

1.3.- OBJETIVOS

El objetivo principal de este Proyecto Fin de Carrera, es el diseño e implementación de diversas funcionalidades que puedan ser integradas en una aplicación Web para un Centro de Salud español, de manera que sea más fácil de administrar dicho centro, y que los datos se encuentren centralizados y accesibles para todos.

Otro de los objetivos de este proyecto es el estudio y posterior uso de nuevas tecnologías, en este caso, Struts e Hibernate, con la finalidad de probarlas y mejorar la implementación, siendo esta una tarea más sencilla y clara.

La especificación de los objetivos es la siguiente:

- Gestión y administración de los datos pertenecientes tanto a los nuevos como antiguos pacientes del Centro de Salud.
- Gestión y administración de los datos de los administradores y médicos del centro.
- Gestión y administración de medicamentos y sus recetas.
- Gestión y administración de información sobre vacunas y edades de suministro.
- Generación y moderación de un foro de consulta para pacientes y médicos.
- Gestión y actualización de información sobre el centro y sus integrantes.
- Creación de un espacio para incluir noticias de interés médico, que será actualizado paulatinamente por los administradores.



- Evaluación automática de los seguimientos de acenocumarol (Sintrom).
- Control de acceso a la aplicación.

1.4.- ESTRUCTURA DE LA MEMORIA

Este documento se encuentra estructurado en los siguientes apartados:

1. **Introducción:** Breve descripción del Proyecto Fin de Carrera y sus objetivos, junto con una descripción de cada uno de los capítulos de este documento.
2. **Estado de la cuestión:** Introducción al contexto del proyecto y descripción de las herramientas y tecnologías usadas para el desarrollo del proyecto.
3. **Gestión del proyecto:** Descripción de la gestión del proyecto incluyendo el ciclo de vida, componentes del proyecto, distribución y planificación de cada una de las tareas que se llevan a cabo para su desarrollo y estimación de costes.
4. **Análisis del sistema:** Representación de los diagramas y modelos realizados para analizar el sistema junto con los diagramas que conforman la base de datos.
5. **Instalación y manual de usuario:** Completa descripción del proceso de instalación de todas las herramientas necesarias para el correcto funcionamiento del proyecto, y elaborado manual de usuario que explica como utilizar de forma eficaz y concisa la aplicación a través de diversos ejemplos e ilustraciones.
6. **Accesibilidad:** Introducción a la accesibilidad en páginas Web y estudio de accesibilidad de algunas páginas de la aplicación a través del programa TAW.
7. **Conclusiones y líneas futuras:** Conclusiones obtenidas tras la realización del proyecto junto con todas aquellas ampliaciones que mejoren las actuales funcionalidades del proyecto.
8. **Bibliografía:** Referencias a las diferentes fuentes de información utilizadas para la elaboración del proyecto.



2.- ESTADO DE LA CUESTIÓN

2.1.- INTRODUCCIÓN

Desde principios de la década de los 80, en España, al igual que en otros países de nuestro entorno, el interés por las redes informáticas fue creciendo, fundamentalmente, en el seno de la comunidad académica y científica. Las grandes necesidades de comunicación de los investigadores dieron lugar, a lo largo de los años 80, a distintas iniciativas que trataban de buscar soluciones particulares a la problemática de determinados colectivos, apoyándose para ello en la existencia de redes homólogas de ámbito internacional.

La iniciativa la tomaron los investigadores españoles en física y altas tecnologías que fueron los precursores de la red FAENET que interconectaba algunas universidades españolas con fines investigadores.

Algunos años antes ya existía una necesidad de disponer de una red informática nacional que sirviera por igual a toda la comunidad académica e investigadora española aunque no se consiguió este propósito hasta bien entrado el año 1987. En estos círculos, la existencia de una red de investigación, en estrecha colaboración con otras redes similares que por esas fechas iban apareciendo en otros países europeos, se consideraba como un instrumento indispensable para el progreso de las diversas disciplinas científicas y tecnológicas.

<i>Julio 1990</i>	Conexión inicial de España a la Internet vía IXI (64 Kbps). Servicio Experimental de RedIRIS.
<i>Diciembre 1990</i>	4 centros conectados experimentalmente: Fundesco RedIRIS), DIT (UPM), CICA y CIEMAT.
<i>Marzo 1991</i>	Inicio fase operativa servicio de acceso al Internet de RedIRIS (SIDERAL). Primeros centros conectados: CIEMAT, CNM, CSIC, Fundesco, RICA, UAB, UAM, UB, UCM, UPC, UPM.
<i>Octubre 1991</i>	Más de 1.000 ordenadores conectados (registrados bajo ".es").
<i>Enero 1992</i>	Primer proveedor comercial: Goya Servicios Telemáticos, S.A.
<i>Abril 1992</i>	RedIRIS participa en la creación del centro de coordinación europeo (RIPE NCC).
<i>Mayo 1992</i>	RedIRIS participa en la creación de Ebone. Conexión inicial a 64 Kbps (posteriormente a 128 Kbps).
<i>Julio 1992</i>	Descentralización del NIC de Internet. ES-NIC (RedIRIS) asume las funciones para España.
<i>Finales 1992</i> <i>Principios 1993</i>	Primeros servidores WWW en España: UNICAN, UJI, UNIOVI
<i>Marzo 1993</i>	Conexión de RedIRIS a MBone (Multicast IP backbone).



<i>Agosto 1993</i>	Más de 10.000 ordenadores conectados (registrados bajo ".es").
<i>Septiembre 1993</i>	Primera conexión internacional 2 Mbps: RedIRIS-EuropaNET.
<i>Mayo 1994</i>	Más de 20.000 máquinas conectadas (registrados bajo ".es"). Más de 100 organizaciones con plena conectividad.
<i>Noviembre 1994</i>	Segundo proveedor comercial: SERVICOM.
<i>Julio 1995</i>	10 proveedores de servicios Internet
<i>Diciembre 1995</i>	30 proveedores de servicios Internet. Lanzamiento de InfoVIA.
<i>Enero 1996</i>	Primera conexión entre proveedores nacionales: RedIRIS-Ibernet.
<i>Octubre 1996</i>	Más de 100.000 ordenadores conectados (registrados bajo ".es"). 200 proveedores de servicios Internet.
<i>Enero 1997</i>	Puesta en marcha de ESPANIX.
<i>Mayo 1997</i>	Conexión de RedIRIS a TEN-34 (22 Mbps). Conexión de RedIRIS al 6bone (IPv6 backbone).
<i>Septiembre 1997</i>	Más de 500 proveedores de servicios Internet. Alrededor de un millón de personas con acceso a Internet.
<i>Septiembre 1997</i>	Más de 200.000 ordenadores conectados (registrados bajo ".es").
<i>Octubre 1998</i>	Más de 270.000 ordenadores conectados (registrados bajo ".es"). Alrededor de 2 millones de personas con acceso a Internet. 35% de empresas españolas con página Web.
<i>Diciembre 1998</i>	Liberalización total de las Telecomunicaciones. Fin de InfoVIA.

- Tabla 1: Cronología de Internet en España -

Tradicionalmente, la expansión de los servicios telemáticos en general y de Internet en particular se ha visto ralentizada en España a causa de los elevados precios de las comunicaciones de larga distancia.

Actualmente España ocupa una posición destacada en el ámbito internacional en cuanto número total de ordenadores conectados a Internet (el noveno país europeo y decimotercero del mundo con más de 270.000), número de organizaciones conectadas o con presencia en la red, número de proveedores de servicios Internet (más de 500) y número de usuarios finales (próximo a los 2 millones de personas, según estimaciones basadas tanto en el número de ordenadores registrados como en diversas encuestas efectuadas entre la población). Nos encontramos en pleno crecimiento tecnológico, cada vez son más las empresas que buscan informatizarse y cada vez existe una mayor demanda de informáticos para cubrir estos trabajos.

España ha tardado más que otros países de la Unión Europea en adaptarse a las nuevas tecnologías, pero en estos momentos se ha disparado notablemente el desarrollo tecnológico que se ha arraigado con gran fortaleza en el sector empresarial a lo largo de estos últimos años, y poco a poco se ha ido consolidando dentro de los mercados internacionales.



2.2.- HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

2.2.1.- INTRODUCCIÓN

La aplicación Web está implementada a través de J2EE conectada a un sistema gestor de Base de Datos como es MySQL utilizando como servidor de aplicaciones Apache Tomcat.

Además de ello, se emplean otras herramientas y tecnologías que se explicarán con más detalle más adelante.

A continuación se muestra un estudio de las principales herramientas que conforman los pilares de este proyecto Struts e Hibernate.

2.2.2.- INTRODUCCIÓN AL MARCO DE TRABAJO STRUTS

2.2.2.1.- INTRODUCCIÓN

La Web ha tenido un largo trayecto, desde sus comienzos en 1984 hasta la actualidad, vamos a hablar un poco de ello.

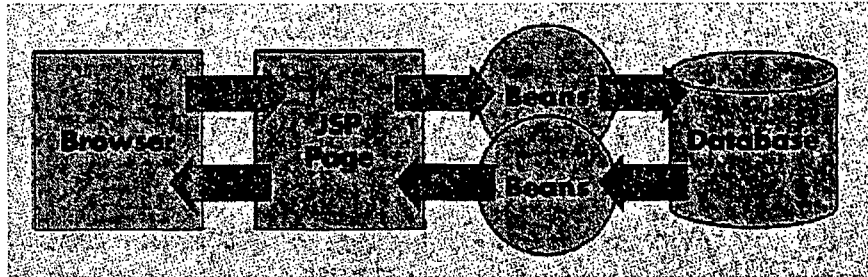
La Web fue creada para intercambiar documentos estáticos aunque finalmente entraron en marcha contenidos dinámicos, todo gracias a la *Common Gateway Interface* (CGI). CGI era un estándar que permitía a los servidores Web interactuar con aplicaciones externas por lo cual las páginas ya no tenían por que ser estáticas. Aunque las aplicaciones CGI fueron muy buenas, contaban con una serie de limitaciones, lo que dio lugar a la creación de los *Servlet* de Java.

Los *Servlet* de Java fueron una revolución ya que constituían una gran mejora frente a la CGI. Los *servlet* proporcionaban un método independiente de la plataforma, basado en componentes para crear aplicaciones Web. Además eran más eficaces que los CGI ya que permitían crear un solo proceso pesado y que múltiples peticiones fueran procesadas en la misma instancia de *servlet*. Su gran problema es que necesitaban una gran interacción con el navegador, ya que por ejemplo había que mandar una gran cantidad de peticiones `println()` al navegador.

Finalmente se comenzaron a desarrollar aplicaciones Web basadas en *Java Server Pages* (JSP), ya que se comprobó que los *servlet* solo tenían incorporado HTML lo cual provocaba un problema de responsabilidad. Las JSP son documentos de textos que combinan etiquetas estáticas HTML y XML además de scripts.

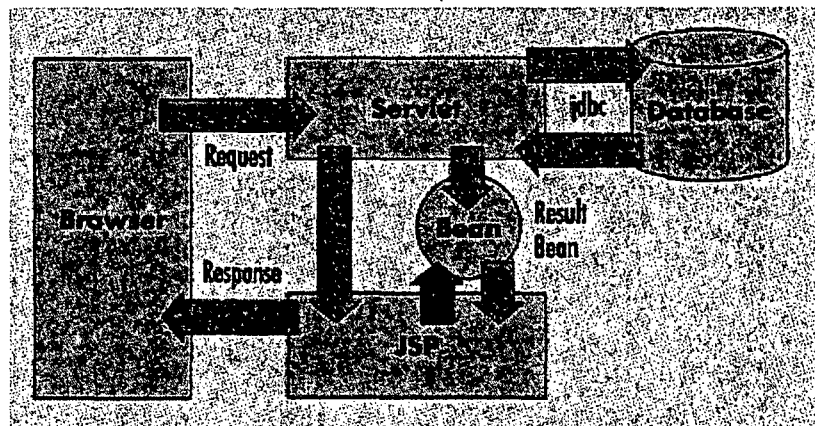
Tras todos estos sucesos y avances de la Web, los desarrolladores se dieron cuenta de que debían combinar las JSP y los *servlets* para desplegar aplicaciones Web. De esta forma aparecieron dos modelos:

- **Modelo 1:** Las páginas *JSP* gestionaban todo el procesamiento de una petición y eran responsables de mostrar el resultado al cliente.



- Figura 1: Modelo1 -

- **Modelo 2:** La petición del cliente la interceptaba un *servlet* controlador que gestionaba el procesamiento inicial de la petición y determinaba que página *JSP* mostrar a continuación.



- Figura 2: Modelo 2 (MVC) -

2.2.2.1.1.- Modelo-Vista-Controlador y Struts

El modelo 2 conocido comúnmente como modelo-vista-controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- El **modelo**, que es el responsable de la lógica de negocio.
- La **vista**, que es la responsable de presentar el contenido.



- El **controlador**, que se encarga de controlar el flujo y estado de la entrada de datos por parte del cliente.

El MVC funciona de la siguiente manera: el flujo de la aplicación está dirigido por un *controlador* central. El *controlador* delega solicitudes (en nuestro caso, solicitudes HTTP) a un manejador apropiado. Los manejadores están unidos a un *modelo*, y cada manejador actúa como un adaptador entre la solicitud y el *modelo*. El *modelo* representa, o encapsula, un estado o lógica de negocio de la aplicación. Luego el control es devuelto a través del *controlador* hacia la *vista* apropiada. El reenvío puede determinarse consultando los conjuntos de mapeos, cargados desde un fichero de configuración en XML. Esto proporciona un acoplamiento cercano entre la *vista* y el *modelo*, que hace que las aplicaciones Web sean más fáciles de crear y de mantener.

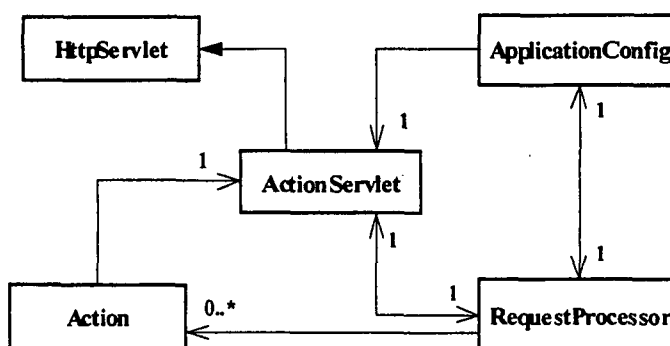
Este modelo dio lugar al nacimiento de *Struts*.

Struts es un marco de trabajo (framework) de código abierto creado por Craig R. McClanahan y donado a la Apache Software Foundation en el año 2000, que proporciona a los desarrolladores una infraestructura unificada sobre la que se pueden basar las aplicaciones Web. A través de este framework se permite que los desarrolladores se centren en la creación de la aplicación de negocio en lugar de las infraestructuras.

2.2.2.2.- CONTROLADOR DE STRUTS

El controlador dentro de una aplicación MVC se encarga de recibir datos de un cliente, invocar una operación de negocio y coordinar la vista a devolver al cliente, por supuesto hay mas acciones, pero estas son las principales.

El marco de trabajo *Struts* utiliza un servlet para procesar las peticiones entrantes, además de muchos otros componentes que vamos a examinar detenidamente.



- Figura 3: Componentes del controlador de Struts -



2.2.2.2.1.- La Clase *ActionServlet*

La clase *org.apache.struts.action.ActionServlet* actúa como un interceptor en las aplicaciones *Struts* y es responsable de empaquetar y enrutar el tráfico HTTP hasta el manejador adecuado.

Cuando una petición del cliente pasa por la *ActionServlet*, se invoca al método *process()* de *ActionServlet* para que la gestione.

Se muestra un ejemplo del método *process()* de *ActionServlet*:

```
protected void process (HttpServletRequest request,
                        HttpServletResponse response)
    throws IOException, ServletException {
    RequestUtils.selectModule (request, getServletContext ());
    getRequestProcessor (getModuleConfig(request)).
    process (request, response);
}
```

2.2.2.2.2.- La Clase *RequestProcessor*

Tras gestionar la petición, el siguiente paso es llamar al método *process()* de la clase *org.apache.struts.action.RequestProcessor*.

Esta clase permite que los desarrolladores personalicen el comportamiento de gestión de petición para una aplicación.

2.2.2.2.3.- La Clase *Action*

La clase *org.apache.struts.action.Action* es el enlace entre una petición del cliente y una operación de negocio. Además también puede realizar funciones de autorización, conexión, validación...

Las clases *Action* llevan a cabo una sola operación de negocio, lo cual no significa que realicen una sola tarea, sino que las tareas que realiza pertenecen a una única funcionalidad.

Esta clase tiene muchos métodos, pero el más importante es el método *execute()*, del cual se muestra un ejemplo:

```
public ActionForward execute (ActionMapping mapping,
                             ActionForm form,
                             HttpServletRequest request,
                             HttpServletResponse response)
    throws Exception;
```



Este método lo invoca la clase *RequestProcessor* cuando recibe una petición del cliente, de forma que el controlador crea una instancia de la clase *Action* si no existe ya alguna.

2.2.2.2.4.- La Clase *ActionForward*

La clase *org.apache.struts.action.ActionForward* representa una abstracción lógica de un recurso Web (normalmente una JSP).

El objeto *ActionForward* contiene detalles que indican donde debe el *ActionServlet* enviar el control para proporcionar una respuesta adecuada.

2.2.2.3.- EL MODELO DE STRUTS

El modelo representa los datos de negocio de una aplicación, por ello se incluyen en el las entidades de negocio y las reglas que gobiernan el acceso y modificación de los datos. Es muy importante que todo ello se mantenga en una única ubicación con el fin de mantener la integridad de los datos, reducir la redundancia y aumentar la reutilidad.

2.2.2.3.1.- Los Objetos de Negocio

Los objetos de negocio que forman parte del modelo, son una abstracción software de una entidad del mundo real. Estos objetos constan de un estado y un comportamiento.

Para que una clase Java sea considerada como un objeto de negocio debe cumplir las siguientes condiciones:

- Tener estado y comportamiento.
- Representar una entidad del mundo real.
- Ser reutilizable.

Los objetos de negocio se pueden agrupar de la siguiente manera:

- ***Objetos de negocio de entidad***, son aquellos que representan una persona, lugar, cosa o concepto. En las aplicaciones Web suelen ser JavaBeans que contienen estado y comportamiento.
- ***Objetos de negocio de proceso***, son aquellos que representan como su nombre indica procesos de negocio dentro de una aplicación. Estos objetos suelen depender de los objetos de entidad ya que son los verbos que los complementan, es decir, son acciones. En las aplicaciones Web suelen ser JavaBeans normales que se comportan como un controlador de la aplicación.



- **Objetos de negocio de evento**, son aquellos que representan algún evento en la aplicación que es generado por alguna acción del sistema.

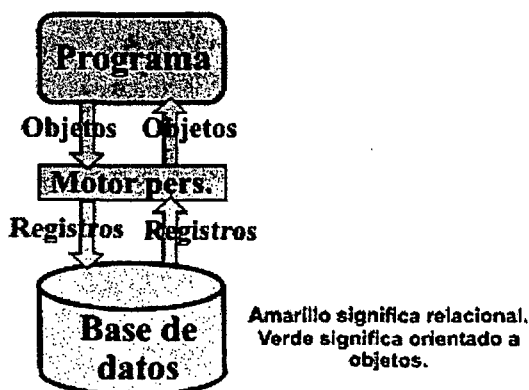
2.2.2.3.2.- La Persistencia

Se entiende por persistencia en la programación orientada a objetos, la capacidad que tienen los objetos de conservar su estado e identidad entre distintas ejecuciones del programa que los creó o de otros programas que accedan a ellos, es decir, que los datos que se incorporan a una aplicación existirán más allá de la vida de esta.

Cuando los objetos se crean en memoria para una aplicación, no pueden estar allí para siempre, con el tiempo tienen que ser eliminados o persistir en un almacén de datos. Una vez que los objetos han persistido, se pueden recuperar y utilizar para reconstruir los objetos de negocio en un futuro.

La persistencia permite al programador almacenar, transferir y recuperar el estado de los objetos. Para esto existen varias técnicas:

- La **serialización** consiste en un proceso de codificación de un objeto en un medio de almacenamiento (como puede ser un archivo, o un buffer de memoria) con el fin de transmitirlo a través de una conexión en red como una serie de bytes o en un formato legible como XML. La serie de bytes o el formato pueden ser usados para re-crear un objeto que es idéntico en su estado interno al objeto original (actualmente un clon). La serialización es un mecanismo ampliamente usado para transportar objetos a través de una red, para hacer persistente un objeto en un archivo o base de datos, o para distribuir objetos idénticos a varias aplicaciones o localizaciones.
- Los **motores de persistencia** traducen los registros a objetos y viceversa. Cuando el programa quiere grabar un objeto llama al motor de persistencia, que traduce el objeto a registros y llama a la base de datos para que guarde estos registros. De la misma manera, cuando el programa quiere recuperar un objeto, la base de datos recupera los registros correspondientes, los cuales son traducidos en formato de objeto por el motor de persistencia.



- Figura 4: Motor de persistencia -



- Las **bases de datos** orientadas a objetos, son aquellas en las que la información se representa mediante objetos como los presentes en la programación orientada a objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS). Un ODBMS hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o más lenguajes de programación a los que dé soporte.

Para crear una aplicación Web utilizando *Struts*, es necesario mapear los objetos de negocio a la base de datos, así según nuestras necesidades podremos usar diversos motores de persistencia, crear uno propio...

2.2.2.3.3.- Patrones Bussiness Delegate y Dao

Al utilizar un motor de persistencia, es necesario desacoplar la persistencia todo lo que sea posible de la aplicación, para ello es necesario crear una interfaz de servicio que pueda utilizar la clase *Action* para que esta clase no interactúe directamente con el marco de persistencia.

El patrón **DAO** se utiliza para desacoplar la lógica de negocio del acceso a datos, además de permitir que la implementación de persistencia sea fácilmente reemplazada por otra sin afectar negativamente a los objetos de negocio.

El patrón **Business Delegate** se encarga de crear y gestionar los objetos de lógica de negocio además de proporcionar un interfaz con los métodos de negocio que se pueden utilizar.

2.2.2.4.- LA VISTA DE STRUTS

Struts utiliza los componentes de vista para mostrar contenido dinámico al cliente. Los principales componentes de vista que se emplean en *Struts* son:

- Documentos HTML.
- Librerías de etiquetas personalizadas JSP.
- JavaScript y hojas de estilo.
- Archivos multimedia.
- Paquetes de recurso de mensaje, los cuales proporcionan un medio para soportar la localización y ayudan a reducir el tiempo de mantenimiento y la redundancia en toda la aplicación.
- Clases *ActionForm*.



2.2.2.4.1.- Uso de JavaBeans

La arquitectura de *JavaBeans* permite a los desarrolladores crear componentes reutilizables que se puedan usar en cualquier plataforma que soporte una máquina virtual de Java. El modelo *JavaBeans* soporta propiedades, eventos, métodos y persistencia.

Dentro de *Struts* solo se utilizan algunas de las posibilidades de *JavaBeans*. Para ello se usa el patrón *Objeto de Transferencia de Datos (DTO)*, el cual agrega varios atributos de una o más entidades de negocio y sitúa esta información en una instancia *JavaBean* mediante la cual se puede recuperar toda la información de un objeto sin sufrir ningún impacto negativo en el rendimiento de la aplicación.

Como resumen, podemos decir que los *DTO* son copias locales de los objetos de negocio que son usadas por la vista para entregar los datos de modelo que se muestran con la información estática en la página.

2.2.2.4.2.- La Clase *ActionForm*

Los objetos de la clase *ActionForm* se utilizan para capturar los datos de entrada de un formulario HTML y transferirlos a la clase *Action*. De esta forma el marco de trabajo recopila automáticamente la entrada de datos de la petición y pasa estos datos a una *Action* utilizando un form bean, que luego pasa a la capa de negocio.

La clase *ActionForm* actúa como un buffer para albergar el estado de los datos que nos ha facilitado el usuario y además actúa como cortafuegos ya que ayuda a mantener la entrada de datos no válida o sospechosa fuera de su nivel de negocio hasta que se hayan analizado las reglas de validación que se pueden crear dentro de esta clase.

Para toda página HTML en la que se utilicen formularios se deberá usar una *ActionForm*, así la misma *ActionForm* se puede utilizar para múltiples páginas si fuera necesario, siempre y cuando coincidan los campos HTML y las propiedades *ActionForm*.

Los *ActionForm* pueden tener dos ámbitos:

- **Petición:** El *ActionForm* solo estará disponible hasta el final del ciclo petición/respuesta. Una vez que se ha devuelto al cliente, la *ActionForm* y los datos dentro de esta dejan de ser accesibles.
- **Sesión:** De esta forma se mantienen los datos del formulario durante más tiempo que solo una petición. Es muy útil cuando por ejemplo creamos un asistente que captura los datos a través de múltiples páginas.

Dentro de una clase *ActionForm* cabe destacar dos métodos:



- **Reset ()** se usa para restablecer las propiedades booleanas de un formulario con el valor false. El controlador llama a este método antes de completar la instancia *ActionForm* con los valores de la petición, así da a *ActionForm* la oportunidad de restablecer sus propiedades a su estado por defecto.
- **Validate ()**, este método devuelve un objeto de la clase *ActionErrors* si se detecta un error ó errores de validación de un formulario, en caso contrario devuelve null. Este método es invocado por el controlador después de que los valores de la petición se hayan insertado en la *ActionForm*.

2.2.2.4.3.- Librerías de Etiquetas

Struts se aprovecha de la característica de librería de etiquetas JSP para incluir varias categorías diferentes de etiquetas que ayudan a hacer la capa de presentación mucho más manejable y reutilizable. Mediante las librerías de etiquetas los desarrolladores pueden interactuar con el resto del marco sin incluir código Java en las páginas JSP.

Las etiquetas JSP personalizadas mantienen su funcionalidad en una clase Java especial denominada el manejador de etiquetas, esta clase tiene acceso a todos los recursos JSP.

Las ventajas de usar etiquetas personalizadas son:

- Las etiquetas son reutilizables, ahorrando por lo tanto tiempo en desarrollo y comprobación.
- Las etiquetas se pueden personalizar utilizando atributos, tanto de forma estática como dinámica.
- Las etiquetas tienen acceso a todos los objetos disponibles para la página JSP, incluidas variables de petición, respuesta y resultado.
- Las etiquetas se pueden anidar, lo que permite interacciones más complejas dentro de una página JSP.
- Las etiquetas simplifican la legibilidad de una página JSP.

Las librerías de etiquetas incluidas con *Struts* son:

- **HTML:** Contiene etiquetas utilizadas para crear formularios de entrada de datos HTML, al igual que otras etiquetas utilizadas generalmente para la creación de interfaces de usuario basadas en HTML. La mayoría de estas etiquetas se debería anidar dentro de una etiqueta de formulario de *Struts*.



Nombre etiqueta	Descripción
<i>button</i>	Muestra un campo de entrada botón.
<i>cancel</i>	Muestra un botón cancelar.
<i>checkbox</i>	Muestra un campo de entrada de datos de casilla de verificación.
<i>errors</i>	Muestra un conjunto de mensajes de error acumulados.
<i>form</i>	Define un elemento de formulario HTML.
<i>hidden</i>	Muestra un campo oculto.
<i>html</i>	Muestra un elemento html de HTML.
<i>image</i>	Muestra una etiqueta de entrada de tipo imagen.
<i>img</i>	Muestra una etiqueta de imagen de HTML.
<i>multibox</i>	Muestra múltiples campos de casilla de verificación.
<i>option</i>	Muestra una opción de selección.
<i>options</i>	Muestra un conjunto de opciones de selección.
<i>optionCollection</i>	Muestra una colección de opciones de colección.
<i>password</i>	Muestra un campo de entrada de contraseña.
<i>radio</i>	Muestra un campo de botón de opción
<i>reset</i>	Muestra un campo de botón restablecer.
<i>submit</i>	Muestra un botón de enviar.
<i>text</i>	Muestra un campo de entrada de datos del tipo texto.
<i>textarea</i>	Muestra un campo de área de texto.

- Tabla 2: Algunas etiquetas de la librería HTML -

- **Bean:** Contiene etiquetas que se utilizan para acceder a JavaBeans y a sus propiedades asociadas, además de para crear nuevos beans.

Nombre etiqueta	Descripción
<i>include</i>	Carga la respuesta desde una petición de aplicación dinámica y la pone disponible como bean.
<i>message</i>	Muestra una cadena de mensaje internacionalizada para la respuesta.
<i>parameter</i>	Define una variable de script basada en el valor del parámetro de petición especificado.
<i>write</i>	Muestra el valor de la propiedad de bean especificada.

- Tabla 3: Algunas etiquetas de la librería Bean -

- **Logic:** Contiene etiquetas que son de utilidad para gestionar la generación condicional de texto de salida, pasar un bucle sobre colecciones de objetos para una generación repetitiva de texto de salida y para la gestión de flujo de la aplicación.



Nombre etiqueta	Descripción
<i>empty</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada es null o cadena vacía.
<i>equal</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada es igual al valor especificado.
<i>forward</i>	Reenvía el control a la página especificada.
<i>greaterEqual</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada es mayor o igual al valor especificado.
<i>greaterThan</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada es mayor que el valor especificado.
<i>iterate</i>	Repite el contenido anidado dentro de ella sobre una colección especificada.
<i>lessEqual</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada es menor o igual que el valor especificado.
<i>lessThan</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada es menor que el valor especificado.
<i>notEmpty</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada no es ni null ni una cadena vacía.
<i>notEqual</i>	Evalúa el contenido anidado dentro de ella si la variable solicitada no es igual que la variable especificada.
<i>redirect</i>	Muestra un redireccionamiento http.

- Tabla 4: Algunas etiquetas de la librería Logic -

- **Tiles:** Son etiquetas que sirven para crear plantillas y evitar repetiremos código en las páginas JSP, dedicaremos un apartado a este tema mas tarde.
- **Nested:** Estas etiquetas sirven para anidar unas etiquetas dentro de otras.

2.2.2.5.- EL MARCO DE TRABAJO VALIDATOR

Validator permite configurar declarativamente rutinas de validación para una aplicación *Struts* sin programar lógica especial de validación.

Aunque ya hemos hablado de la posibilidad de añadir lógica de validación dentro de la clase *ActionForm*, existen varios problemas con este enfoque:

1. Codificar la lógica de validación dentro de cada *ActionForm* provoca redundancia en toda la aplicación.
2. Si se necesita modificar o mejorar la validación que ocurre en un *ActionForm*, el código fuente se tiene que volver a compilar, por ello esto provoca un problema de rendimiento.

Validator permite mover toda la lógica de validación fuera de la *ActionForm* y configurarla de forma declarativa para una aplicación por medio de archivos XML. Por



otro lado, es muy extensible ya que aunque proporciona algunas rutinas de validación, este marco se puede ampliar fácilmente.

2.2.2.5.1.- El Archivo de Validation-Rules.xml

Este archivo de configuración contiene un conjunto global de reglas que cualquier aplicación puede configurar manualmente. Este archivo solo tendrá que ser modificado si se desea ampliar o modificar las reglas que hay en él.

El elemento *validator* soporta siete atributos:

- **name:** Asigna un nombre lógico a la regla de validación. El nombre debe ser único.
- **classname:** Define la clase que contiene la lógica para la regla de validación.
- **method:** Define el método que contiene la lógica para la regla de validación.
- **methodParams:** Es una lista delimitada por comas, que contiene los parámetros para el método definido en el atributo *method*.
- **msg:** Es una clave del paquete de recursos. *Validator* utiliza este valor para buscar un mensaje del paquete de recursos de Struts cuando ocurre un error de validación.
- **depends:** Se utiliza para especificar otras reglas de validación que se deberían invocar antes de la regla que lo especifica.
- **jsFunctionName:** Este atributo opcional, permite especificar el nombre de la función JavaScript que utiliza la regla de validación.

Nombre del método	Descripción
isBlankOrNull	Comprueba si el campo no está vacío y la longitud del campo es mayor que cero.
isByte	Comprueba si el valor se puede convertir a un primitivo del tipo byte.
isCreditCard	Comprueba si el campo es un número de tarjeta de crédito válido.
isDate	Comprueba si el campo es una fecha válida.
isDouble	Comprueba si el valor se puede convertir a un primitivo de tipo double.
isEmail	Comprueba si el campo es una dirección de correo electrónico válida.
isFloat	Comprueba si el valor se puede convertir a un primitivo de tipo flota.



isInRange	Comprueba si el valor está dentro de un rango mínimo y máximo.
isInt	Comprueba si el valor se puede convertir a un primitivo de tipo int.
isShort	Comprueba si el valor se puede convertir a un primitivo de tipo short.
matchRegex p	Comprueba si el valor coincide con la expresión regular.
maxLength	Comprueba si la longitud del valor es menor que o igual al máximo.
minLength	Comprueba si la longitud del valor es mayor que o igual al mínimo.

- Tabla 5: Reglas de validación en la clase GenericValidator -

2.2.2.5.2.- El Archivo Validation.xml

Este archivo es específico de la aplicación; describe qué reglas de validación del archivo validation-rules.xml utiliza una *ActionForm* determinada.

En este archivo el elemento más exterior es *form-validation* que puede contener dos elementos hijos:

- El elemento global, que permite configurar elementos constantes para todo el archivo, como por ejemplo:

```
<global>
  <constant>
    <constant-name> Telefono >/constant-name>
    <constant-value>^\(?\d{3})\)?[-|]?(\d{3})[-|]?(\d{4})$
    </constant-value>
  </constant>
</global>
```

- El elemento formset, que soporta dos atributos que tratan con I18N.

2.2.2.6.- EL MARCO DE TRABAJO TILES

El entorno de trabajo *Tiles* proporciona un mecanismo de plantillas mediante el cual separar las responsabilidades de diseño de las de contenido, además proporciona las siguientes características:

- Posibilidades de uso de plantillas.
- Construcción y carga de página dinámica.
- Definiciones de pantalla.
- Soporte para reutilización de página y diseño.

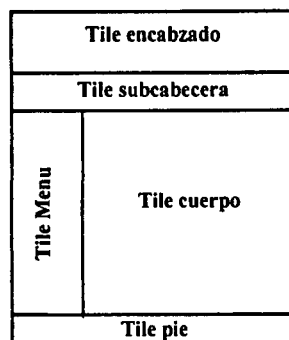


- Soporte de internacionalización.
- Soporte de múltiples canales.

Antes de continuar vamos a explicar que es una plantilla, para entender mejor Tiles.

Una plantilla es una página JSP que utiliza una librería de etiquetas personalizada JSP para describir el diseño de una página. La plantilla actúa como una definición del aspecto que tendrán las páginas de una aplicación, sin especificar el contenido. El contenido se inserta en la página de plantilla en el momento de ejecución. De esta forma, una o más páginas pueden utilizar la misma plantilla.

Un *tile* es un área de trabajo o región dentro de una página Web. Una página puede contar con una o varias regiones.



- Figura 5: Ejemplo de tiles de una aplicación Web -

2.2.2.6.1.- La Librería de Etiquetas Tiles

Como ya se especifico antes, *tiles* es una de las librerías de etiquetas personalizadas de *Struts*, ahora es el momento de explicar más detalladamente las etiquetas que componen esta librería.

Nombre de etiqueta	Descripción
add	Añade un elemento a las lista.
definition	Crea una definición de componente tile.
get	Obtiene el contenido del ámbito de petición que sitúo allí una etiqueta put.
getAsString	Muestra el valor del atributo de tile especificado para el JspWriter actual.
importAttribute	Importa un atributo de tile en el contexto especificado.
initComponentDefinitions	Inicializa un factory de definiciones tile.
insert	Inserta un componente tile.
put	Sitúa un atributo en un contexto de tile.
putList	Declara una lista que se pasará como un atributo.
useAttribute	Utiliza un valor de atributo dentro de una página.

- Tabla 6: Etiquetas de la librería de etiquetas Tiles -



2.2.2.6.2.- Uso de Definiciones

La mayoría de las veces, los atributos que usaremos en cada *tile* serán constantes en toda la aplicación, por ello será redundante especificarlos en cada *tile*. Lo más común es declarar los atributos en un solo lugar, y que los *tiles* los incluyeran allí donde los necesiten.

Las definiciones de *tiles* proporcionan esta funcionalidad. Una definición permite especificar estáticamente los atributos que utiliza una plantilla, lo que a su vez permite, especificar solamente los atributos específicos de página en los *tiles*. Las definiciones permiten:

- Centralizar la declaración de descripción de página.
- Evitar la declaración repetitiva de páginas casi idénticas.
- Especificar el nombre de una definición como un forward en el archivo de configuración de Struts.
- ...

Las definiciones se pueden declarar en una JSP o en un archivo XML.

2.2.2.7.- CONFIGURAR APLICACIONES STRUTS

Struts utiliza dos tipos de archivos de configuración distintos, que se deben configurar adecuadamente antes de que una aplicación funcione correctamente. Ambos archivos están basados en XML.

El primero de ellos es el descriptor de despliegue de la aplicación Web denominado *web.xml*. Este archivo es necesario para cualquier aplicación Web aunque no haya sido creada con *Struts*, pero en él existe información de despliegue específica de *Struts* que se debe configurar cuando se crea una aplicación de este tipo.

El segundo archivo es el de configuración de *Struts*, conocido generalmente como *struts-config.xml*. En él se puede configurar de forma declarativa muchos de sus parámetros de aplicación.

2.2.2.7.1.- Configurar el Archivo Web.xml para Struts

➤ Mapear la ActionServlet

Primero se debe configurar la *ActionServlet* que recibirá todas las peticiones entrantes para la aplicación. Hay que realizar dos pasos para configurarla, el primer paso es utilizar el elemento *servlet* para configurar la instancia de *servlet* que se puede mapear más tarde en el elemento *servlet-mapping*.

Algunos elementos hijos de *servlet* son:



Nombre elemento	Descripción
<i>servlet-name</i>	Especifica el nombre usado por el descriptor de despliegue para hacer referencia al servlet durante todo el archivo.
<i>servlet-class</i>	Indica la clase a la que pertenece el servlet.
<i>init-param</i>	Especifica los parámetros de configuración del servlet.

- Tabla 7: Elementos hijos del elemento servlet -

El siguiente paso es configurar el mapeado del servlet utilizando el elemento *servlet-mapping*.

Se muestra un ejemplo de configuración del servlet:

```
<servlet>
  <servlet-name>nombreDelServlet</servlet-name>
  <servlet-class>org.apache.struts.action.ActionServlet
</servlet-class>
  <init-param>
    <param-name>host</param-name>
    <param-value>localhost</param-value>
  </init-param>
  <init-param>
    <param-name>port</param-name>
    <param-value>7001</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>nombreDelServlet</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

➤ Configurar las librerías de etiquetas

El entorno de trabajo Struts proporciona varias librerías de etiquetas JSP que deben ser configuradas en el descriptor de despliegue de la aplicación para poder utilizarlas.

Para declarar una librería de etiquetas debemos utilizar el elemento *taglib*, el cual contiene dos elementos:

Nombre elemento	Descripción
<i>taglib-uri</i>	Especifica una URI que identifica una librería de etiquetas.
<i>taglib-location</i>	Especifica la ubicación del archivo descriptor de librería de etiquetas.

- Tabla 8: Elementos hijos del elemento taglib -



Se muestra un ejemplo de configuración de las librerías de Struts:

```
<taglib>
    <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-logic.tld
    </taglib-location>
</taglib>
```

➤ Establecer la lista de archivo de bienvenida

El elemento *welcome-file-list* permite configurar recursos por defecto que se deberían usar cuando se facilita un URI válido pero parcial.

Se muestra un ejemplo de configuración de *welcome-file-list*:

```
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

➤ Configurar gestión de error

Aunque el marco de desarrollo de *Struts* proporciona un mecanismo adecuado de gestión de error, en algunos momentos se pueden producir excepciones servlet o JSP que se envían al usuario. Para impedir que esto ocurra, debemos utilizar el elemento *error-page*.

Si se desea controlar una excepción JSP debemos indicar el código del error y la página a la que queremos dirigirnos cuando ocurra, por ejemplo:

```
<error-page>
    <error-code>404</error-code>
    <location>/common/404.jsp</location>
</error-page>
```

En cambio si se desea controlar una excepción de servlet, debemos indicar el tipo de excepción (*RuntimeException*, *ServletException* ó *IOException*) y la página a la que queremos dirigirnos cuando ocurra, por ejemplo:

```
<error-page>
    <error-type>javax.servlet.ServletException</error-type>
    <location>/common/system-error.jsp</location>
</error-page>
```




2.2.2.7.2.- Configurar el Archivo Struts-Config.xml

➤ El elemento *data-sources*

El elemento *data-sources* permite establecer un origen de datos rudimentario que se puede utilizar dentro de *Struts*. Un origen de datos actúa como una fábrica para conexiones de BD y proporciona un único punto de control.

Este elemento puede contener cero o más elementos *data-source*, y cada elemento *data-source* puede contener multiples elemento *set-property* medianotes los cuales se especifica el nombre y valor de una propiedad de configuración JavaBeans adicional cuyo método *setter* se llamará en el objeto que representa el elemento de alrededor.

Nombre elemento	Descripción
<i>className</i>	La clase de implementación del bean de configuración que albergará la información de origen de datos.
<i>key</i>	El atributo de contexto de servlet debajo del que se almacenará este origen de datos.
<i>type</i>	El nombre de la clase Java que implementa el origen de datos.

- Tabla 9: Atributos para el elemento *data-source* -

Se muestra un ejemplo de configuración para conectarse a una BD MySQL:

```
<data-sources>
  <data-source>
    <set-property property="autoCommit" value="true"/>
    <set-property property="description" value="MySQL
    Data Source"/>
    <set-property property="driverClass"
    value="com.jdbc.mysql.Driver"/>
    <set-property property="url"
    value="jdbc:mysql://localhost:3360/Ejemplo"/>
  </data-source>
</data-sources>
```

➤ El elemento *form-beans*

El elemento *form-beans* permite configurar múltiples clases *ActionForm*. Dentro de este elemento se pueden configurar cero o más elementos *form-bean*.



Nombre elemento	Descripción
<i>className</i>	La clase de implementación del bean de configuración que albergará la información del form-bean.
<i>dynamic</i>	Si estamos utilizando Action dinámicos esta propiedad contendrá el valor true.
<i>name</i>	Identificador único para el bean durante todo marco.
<i>type</i>	El nombre de la clase Java que implementa el form-bean.

- Tabla 10: Atributos del elemento form-bean -

Se muestra un ejemplo de configuración de *form-beans*:

```
<form-beans>
  <form-bean
    name="tablasAuxiliaresForm"
    type="com.oreilly.struts.tablasauxiliares.TablasAuxiliaresForm"/>
</form-beans>
```

➤ El elemento global-exceptions

Aquí se pueden configurar manejadores de excepción de forma declarativa.

Nombre elemento	Descripción
<i>className</i>	La clase de implementación del bean de configuración que albergará la información de excepción.
<i>handler</i>	El nombre de la clase Java completo del manejador de excepción que gestionará la excepción.
<i>key</i>	Una clave de mensaje que se especifica en el paquete de recursos para este módulo.
<i>path</i>	La ruta de acceso relativa a la que de producirá el reenvío en caso de excepción.
<i>scope</i>	Ámbito de almacenamiento de la instancia ActionError.
<i>type</i>	El nombre de la clase Java que implementa la excepción.
<i>bundle</i>	El atributo ServletContext que identifica el paquete de recursos donde debería recoger la clave.

- Tabla 11: Atributos del elemento exception -

Se muestra un ejemplo de configuración de *global-exceptions*:

```
<global-exceptions>
  <exception
    key="global.error.invalidLogin"
    path="/security/signin.jsp"
    scope="request"
```



```
type="com.oreilly.struts.framework.exceptions.  
InvalidLoginException"/>  
</global-exceptions>
```

➤ El elemento *global-forwards*

Toda acción que se ejecuta termina al reenviarse o redirigirse a una vista que suele ser una página JSP o HTML, así en vez de hacer referencia directamente a la vista, *Struts* utiliza el concepto de reenvío para asociar un nombre lógico al recurso.

El elemento *global-forwards* permite configurar reenvíos que pueden ser usados por las acciones de la aplicación.

Nombre elemento	Descripción
<i>className</i>	La clase de implementación del bean de configuración que albergará la información de reenvío.
<i>contextRelative</i>	Este atributo se pondrá a true cuando el elemento path sea la ruta relativa.
<i>name</i>	Identificador único para hacer referencia al reenvío durante todo marco.
<i>path</i>	La ruta de acceso a la que de producirá el reenvío.
<i>redirect</i>	Valor booleano que indica si es un reenvío (false) o una redirección (true).

- Tabla 12: Atributos del elemento forward -

Se muestra un ejemplo de configuración de *global-forwards*:

```
<global-forwards>  
  <forward name="operacionCorrecto"  
    path="/comun/operacionCorrecto.jsp" redirect="true"/>  
  <forward name="operacionError"  
    path="/comun/operacionError.jsp" redirect="true"/>  
</global-forwards>
```

➤ El elemento *action-mappings*

Este elemento contiene un conjunto de cero o más elementos *action* para una aplicación Struts.

El elemento *action* describe un mapeado desde una ruta de petición específica a una clase *Action* correspondiente.



Nombre elemento	Descripción
<i>className</i>	La clase de implementación del bean de configuración que albergará la información de acción.
<i>forward</i>	La ruta de acceso relativa a la aplicación para un servlet o recurso JSP al que se reenviará.
<i>name</i>	El nombre del form bean asociado a esta acción.
<i>path</i>	La ruta de acceso relativa a la aplicación para la petición enviada.
<i>scope</i>	Ámbito en el que se sitúa el form bean.
<i>type</i>	El nombre de la clase Java que implementa acción.
<i>validate</i>	Valor booleano que indica si el método validate() del form bean se debe llamar.

- Tabla 13: Algunos atributos del elemento action. -

Se muestra un ejemplo de configuración de *action-mappings*:

```
<action
    path="/buscarTablasAuxiliares"
    type="com.oreilly.struts.tablasauxiliares.
    BuscarTablasAction"
    name="tablasAuxiliaresForm"
    scope="request"
    validate="false">
    <forward name="listaTablas"
    path="/jsp/tablasAuxiliares/buscarTablas.jsp"
    redirect="false"/>
</action>
```

➤ El elemento message-resources

Este elemento especifica las características del paquete de recursos de mensajes que contiene los mensajes localizados para una aplicación.

Nombre elemento	Descripción
<i>className</i>	La clase de implementación del bean de configuración que albergará la información de message-resources.
<i>factory</i>	Nombre de la clase Java de la clase MessageResourcesFactory que se debería usar.
<i>key</i>	El atributo de contexto del servlet con el que se almacenará el paquete de recurso de mensaje.
<i>null</i>	Valor booleano que indica como la subclase MessageResources debería gestionar el caso cuando se use una clave de mensaje desconocida.
<i>parameter</i>	El nombre base del paquete de recursos.

- Tabla 14: Atributos del elemento message-resources -



Se muestra un ejemplo de configuración de *message-resources*:

```
<message-resources  
parameter="com.oreilly.struts.resources.ApplicationResources"/>
```

➤ El elemento *plug-in*

Esta potente característica permite que las aplicaciones *Struts* descubran recursos dinámicamente cuando se inician.

Para crear un *plug-in*, se necesita crear una clase Java que implemente la interfaz *org.apache.struts.action.plugin* y añadir un elemento *plug-in* al archivo de configuración.

Se muestra un ejemplo de la configuración del *plug-in* para el framework *Validator*:

```
<plug-in  
  className="org.apache.struts.validator.ValidatorPlugIn">  
  <set-property  
    property="pathnames"  
    value="/WEB-INF/validator-rules.xml,  
          /WEB-INF/validator-rules-custom.xml,  
          /WEB-INF/validation.xml"/>  
</plug-in>
```

2.2.3.- INTRODUCCIÓN AL MARCO DE TRABAJO HIBERNATE

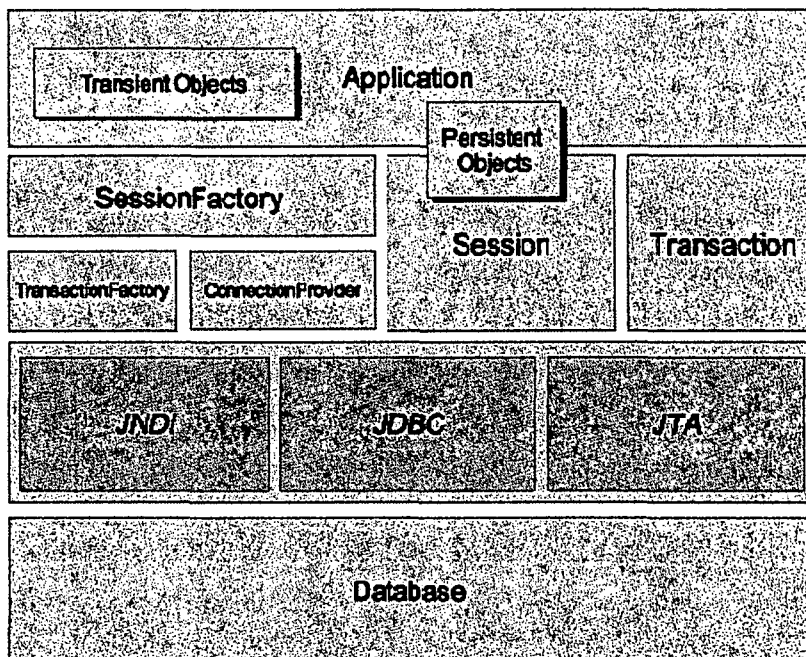
2.2.3.1.- INTRODUCCIÓN

Para la mayoría de las aplicaciones, almacenar y recuperar información implica alguna forma de interacción con una base de datos relacional. Esto ha representado un problema fundamental para los desarrolladores ya que algunas veces el diseño de datos relacionales y los ejemplares orientados a objetos comparten estructuras de relaciones muy diferentes dentro de sus respectivos entornos.

Las bases de datos relacionales están estructuradas en una configuración tabular y los ejemplares orientados a objetos normalmente están relacionados en forma de árbol. Esta diferencia es lo que ha provocado que se desarrollen diversas tecnologías de persistencia de objetos intentando así construir un puente entre el mundo relacional y el mundo orientado a objetos.

Hibernate es un marco de trabajo que tiene por objetivo facilitar la persistencia de objetos Java en bases de datos relacionales y al mismo tiempo la consulta de estas bases de datos para obtener objetos, es decir, que *Hibernate* es una herramienta que realiza el mapeado entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM

(object/relational mapping) y consiste en la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.



- Figura 6: Arquitectura de Hibernate -

2.2.3.2.- CONCEPTOS BÁSICOS DE HIBERNATE

Hibernate funciona asociando a cada tabla de la base de datos un Plain Old Java Object (POJO). Un POJO es similar a una Java Bean, con propiedades accesibles mediante métodos setter y setter.

Para poder asociar el POJO a su tabla correspondiente en la base de datos, *Hibernate* usa los ficheros de configuración de mapeo hbm.xml. En estos ficheros se declaran las propiedades del POJO y sus correspondientes nombres de columna en la base de datos, asociación de tipos de datos, referencias, relaciones x a x con otras tablas...

2.2.3.3.- EL FICHERO DE CONFIGURACIÓN DE HIBERNATE

El fichero *hibernate.cfg.xml* es en el cual se determinan los aspectos relacionados con el gestor de base de datos y las conexiones a él.

Se muestra un ejemplo de fichero de configuración de *Hibernate*:



```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-
2.0.dtd">

<hibernate-configuration>

<session-factory>
  <property name="connection.driver_class">
    com.mysql.jdbc.Driver</property>
  <property name="connection.url">
    jdbc:mysql://localhost/test</property>
  <property name="connection.username">yankoa</property>
  <property name="connection.password"></property>
  <property name="dialect">
    net.sf.hibernate.dialect.MySQLDialect</property>
  <property name="show_sql">true</property>
  <property name="transaction.factory_class">
    net.sf.hibernate.transaction.JDBCTransactionFactory
  </property>

  <!-- Mapping files -->
  <mapping resource="com/shop/Category.hbm.xml" />
  <mapping resource="com/shop/Product.hbm.xml" />
</session-factory>

</hibernate-configuration>
```

Los cuatro primeros *property* contiene la configuración necesaria para la conexión JDBC que utilizara *Hibernate*, es decir, se indica la clase que contiene el driver para conectarse a la base de datos, url de la conexión y el nombre y contraseña para acceder a la base de datos.

Mediante la propiedad *dialect* se indica el nombre de la clase que se encargará de comunicarse con la base de datos en el lenguaje SQL que entienda la base de datos. Este parámetro ha de ser siempre especificado y el valor ha de ser una subclase que herede de *net.sf.hibernate.dialect.Dialect*.

Hibernate proporciona los siguientes dialectos:

RDBMS	Dialect
DB2	net.sf.hibernate.dialect.DB2Dialect
MySQL	net.sf.hibernate.dialect.MySQLDialect
SAP DB	net.sf.hibernate.dialect.SAPDBDialect
Oracle (any version)	net.sf.hibernate.dialect.OracleDialect
Oracle 9	net.sf.hibernate.dialect.Oracle9Dialect
Sybase	net.sf.hibernate.dialect.SybaseDialect
Sybase Anywhere	net.sf.hibernate.dialect.SybaseAnywhereDialect
Progress	net.sf.hibernate.dialect.ProgressDialect
Mckoi SQL	net.sf.hibernate.dialect.MckoiDialect



RDBMS	Dialect
Interbase	net.sf.hibernate.dialect.InterbaseDialect
Pointbase	net.sf.hibernate.dialect.PointbaseDialect
PostgreSQL	net.sf.hibernate.dialect.PostgreSQLDialect
HypersonicSQL	net.sf.hibernate.dialect.HSQLDialect
Microsoft SQL Server	net.sf.hibernate.dialect.SQLServerDialect
Ingres	net.sf.hibernate.dialect.IngresDialect
Informix	net.sf.hibernate.dialect.InformixDialect
FrontBase	net.sf.hibernate.dialect.FrontbaseDialect

- Tabla 15: Dialectos de Hibernate -

En los siguientes *property* se indica que *Hibernate* delegará las transacciones a la conexión JDBC subyacente y después especificamos el proveedor de caché.

Finalmente se indica a través de *mapping-resource* el path donde se encuentran los ficheros de mapeo.

2.2.3.4.- EL FICHERO DE MAPEO DE HIBERNATE

Para poder asociar el POJO a su tabla correspondiente en la base de datos, *Hibernate* usa los ficheros de mapeo con el sufijo hbm.xml.

En estos ficheros se declaran las propiedades del POJO y sus correspondientes nombres de columna en la base de datos, asociaciones de tipos de datos, referencias, relaciones x a x con otras tablas...

Se muestra un ejemplo de un fichero de mapeo de *Hibernate*:

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-
    2.0.dtd">

<hibernate-mapping>
    <class name="escuela.beans.Categoria" table="categorias">
        <id name="id" type="integer" column="ID" unsaved-
            value="-1">
            <generator class="identity"/>
        </id>

        <property name="categoria" column="CATEGORIA"
            type="string"
            unique="true"
            not-null="true"/>
    </class>
</hibernate-mapping>
```




2.2.3.4.1.- El Elemento <hibernate-mapping>

Este elemento permite especificar, mediante atributos, diversas características como el esquema de la base de datos.

2.2.3.4.2.- El Elemento <class>

Mediante este elemento especificamos el emparejamiento entre una clase java con una tabla de la base de datos.

A través de *name*, especificamos el nombre de la clase Java, y mediante *table*, el nombre de la tabla de la base de datos con la que se empareja.

2.2.3.4.3.- Los Elementos <id> y <property>

Una vez emparejados objeto y tabla, se necesita emparejar las propiedades del objeto con las columnas de la tabla, para ello se usan dos elementos:

- **<id>**: Empareja una de las propiedades del objeto con la clave primaria de la tabla. A través de *name*, identificamos la propiedad del objeto que se empareja; con *column*, indicamos con que columna de la tabla se empareja; opcionalmente con *type* indicamos el tipo de datos con el que estamos trabajando y mediante *unsaved-value*, opcionalmente indicamos el valor que debe tener el campo clave cuando todavía no ha sido almacenado el objeto en la base de datos.
- **<property>**: Empareja una de las propiedades del objeto con algún atributo de la tabla, que no sea clave primaria. Es muy similar a <id> pero tiene dos nuevas propiedades: *unique* que permite especificar si se admiten o no valores duplicados (es opcional) y *not-null* que indica si puede ser una propiedad nula o no.

2.2.3.4.4.- El Subelemento <generator>

Este elemento permite definir cómo se generan los valores de las claves primarias, existen diversos métodos para ello, pero solo se comentarán los más utilizados:

- **identity**: Utiliza el soporte de campos *identity* que ofrecen la mayoría de las bases de datos para generar la clave automáticamente.
- **assigned**: Permite que sea la propia aplicación quien elija un identificador para el objeto antes de almacenarlo en la base de datos.
- **sequence**: Utiliza una secuencia para generar la clave.



- **hilo:** Utiliza una tabla auxiliar para generar las claves.
- **native:** Permite que *Hibernate* escoja entre los métodos *identity*, *sequence* o *hilo* en función de las características del gestor de base de datos con el que se trabaje.

2.2.3.4.5.- Mapear Asociaciones

Se ha mostrado como mapear un único objeto a la base de datos, es decir, relaciones uno a uno 1-1, pero en todo diseño relacional los objetos se referencian unos a otros a través de relaciones mas complejas como son : uno a muchos 1-n, muchos a muchos n-m, muchos a uno n-1.

Para poder explicar las asociaciones se utilizaran diversos ejemplos, con el fin de dejar más claro su uso.

➤ Asociaciones n-1

La relación “*muchos a uno*” necesita en la tabla un identificador de referencia, el ejemplo clásico es la relación entre padre - hijos. Un hijo necesita un identificador en su tabla para indicar cual es su padre. Pero en objetos en realidad no es un identificador si no el propio objeto padre, por lo tanto el componente n-1 es en realidad el propio objeto padre y no simplemente su identificador (aunque en la tabla se guarde el identificador).

Para reflejar este tipo de relación se utiliza el elemento `<many-to-one>` que contiene los siguientes atributos:

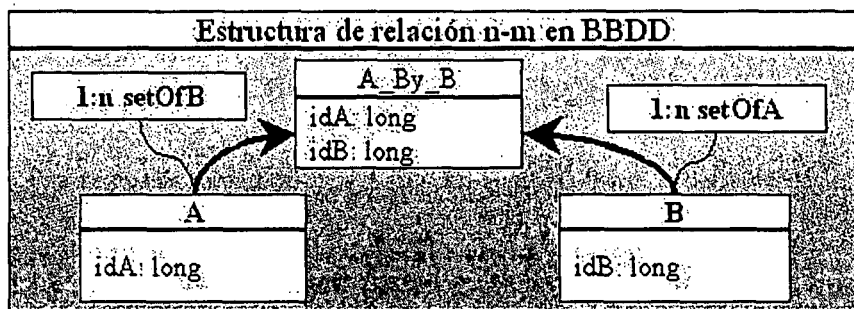
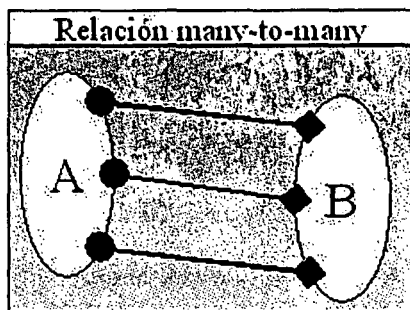
- **name:** El nombre de la propiedad.
- **column:** Columna de la tabla donde se guardara el identificador del objeto asociado.
- **class:** Nombre de la clase asociada.
- **cascade (“all|none|save-update|delete”):** Especifica que operaciones se realizaran en cascada desde el objeto padre. Este atributo es opcional.

Se muestra un ejemplo de una relación n-1:

```
<many-to-one name="padre" column="ID_PADRE" class="padreR"/>
```

➤ Asociaciones n-m

En las asociaciones “muchos a muchos” se contará con dos clases A y B, donde un elemento de la clase A tendrá un conjunto de hijos de la clase B, y un elemento de B tendrá otro conjunto igual o distinto de elemento A.



- Figura 7: Relación n-m -

Para reflejar este tipo de relación se utiliza el elemento *<many-to-many>* que contiene los siguientes atributos:

- **name:** El nombre de la propiedad.
- **table:** Nombre de la tabla de la colección.
- **column:** Columna de la tabla donde se guardara el identificador del objeto asociado.
- **class:** Nombre de la clase asociada.
- **key:** Columna clave que se relaciona con la clave primaria del objeto intermedio.
- **lazy ("true"|"false"):** Permite el uso de inicialización "lazy". Este tipo de inicialización hace que los objetos de la colección sean solicitados en demanda y no se carguen todos a la vez.
- **inverse:** Señala la colección como el fin de una asociación bidireccional.
- **cascade:** Permite las operaciones en cascada hacia los entidades hijas. Puede contener los siguientes valores:
 - **save-update:** Se aplica solo a las operaciones de inserción o modificación.
 - **delete:** Se aplica solo a las operaciones de borrado.
 - **all:** Se aplica a cualquier operación.
 - **none:** No se realiza ninguna operación en cascada.
- **sort:** Especifica una colección con una ordenación natural o con una clase comparadora dada.
- **order-by:** Columnas de la tabla que definen el orden de iteración. Puede ser ascendente o descendente.



Para diseñar este tipo de asociaciones, se necesita crear una tabla intermedia que relacione los códigos de los elementos de A con los elementos de B. Por lo tanto se ha de mapear en una tabla a parte con las claves de las dos tablas como claves ajenas.

Esto se podría mapear de la siguiente manera:

```
<set role="setOfB" table="A_By_B">  
  <key column="A_id"/>  
  <many-to-many column="B_id" class="elementOfB"/>  
</set>
```

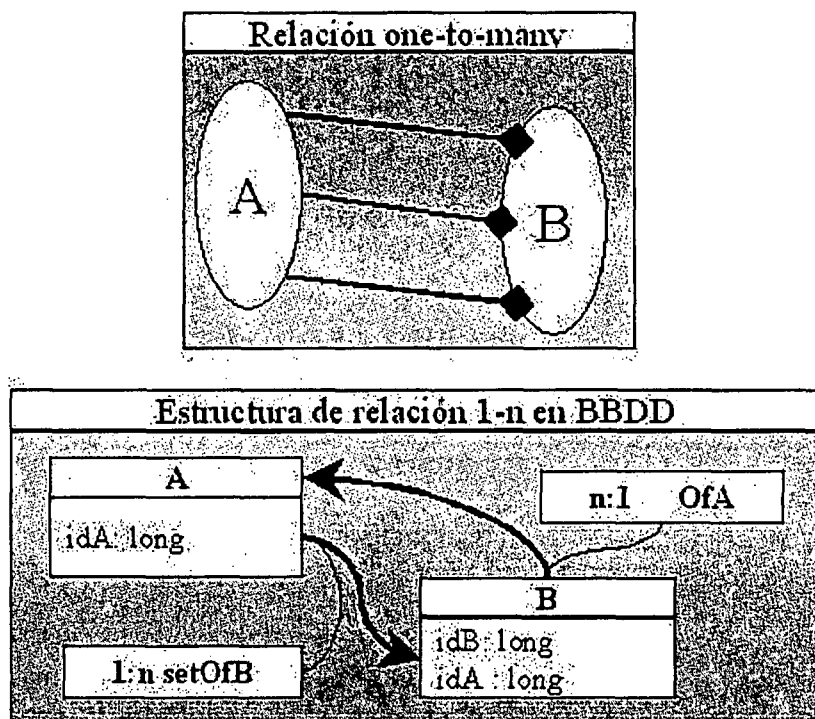
Como se puede observar, no existe una columna extra en B que diga los elementos de B que le corresponden a un elemento de A. En vez de eso se ha creado una tabla nueva A_By_B que contiene los pares de claves relacionados tanto de A hacia B como de B hacia A. Para que sea bidireccional tiene que ser declara en el mapeo de la clase B como sigue.

```
<set role="setOfA" table="A_By_B" inverse="true">  
  <key column="B_id"/>  
  <many-to-many column="A_id" class="elementOfA"/>  
</set>
```

➤ Asociaciones 1-n

Para crear relaciones “uno a muchos”, es necesario mapear correctamente las dos tablas que participan en la relación, creando por un lado la relación <one-to-many> y por otro una relación <many-to-one>.

Este tipo de relación puede contener los mismos atributos que las relaciones <many-to-many>.



- Figura 8: Relación 1-n -

Una asociación 1-n de A hacia B requerirá un nuevo campo en B con el valor del índice de A al que se encuentra asociado. En la tabla A no será necesario ningún nuevo campo.

Se muestra un ejemplo de relación 1-n:

```
<set name="setOfB" table="B">
  <key column="A_id"/>
  <one-to-many class="B"/>
</set>
```

2.2.3.5.- LAS SESIONES

Para que se puedan utilizar los mecanismos de persistencia de *Hibernate*, el desarrollador debe mantener una conversación con el motor de *Hibernate* mediante un objeto especial, el objeto de "sesión".

Se puede equiparar a grandes rasgos al concepto de conexión de JDBC y cumple un papel muy parecido, es decir, sirve para delimitar una o varias operaciones relacionadas dentro de un proceso de negocio, demarcar una transacción y además aporta algunos servicios adicionales como una caché de objetos para evitar interacciones innecesarias contra la BD. En este sentido, la clase *Session* ofrece métodos como *save(Object object)*, *createQuery(String queryString)*, *beginTransaction()*, *close()*, para interactuar con la BD tal y como se hace con una conexión JDBC con la



diferencia de que existe una mayor simplicidad, es decir, guardar un objeto, por ejemplo, consiste en algo así como *session.save(miObjeto)*, sin necesidad de especificar una sentencia SQL.

Las sesiones son un concepto ligado a un proceso de negocio, por tanto es natural pensar que una sesión siempre va a pertenecer a un mismo thread de ejecución, aunque técnicamente se pueden compartir sesiones entre threads, esto no se debe hacer jamás por no ser una buena política de diseño y los consecuentes problemas que puede generar, es decir, en un entorno multiusuario y por tanto multithread habrá múltiples sesiones simultáneas, cada una perteneciente a su correspondientes thread y con su contexto de objetos en caché, transacciones, etc. Como tal no sorprende que las sesiones no son “thread-safe” y que la información vinculada a ella no sea visible para otras sesiones. Es también lógico que tenga que existir una “institución” superior para crear sesiones y realizar operaciones comunes a los diferentes threads como lo puede ser la gestión de una caché compartida entre threads o caché de segundo nivel. Este elemento es la clase *SessionFactory* y en ella podremos encontrar métodos como *openSession()* o *evict(Class persistentClass)*.

2.2.3.6.- HIBERNATE QUERY LENGUAJE: HQL

Hibernate ofrece un lenguaje de consultas que agrupa un potente y flexible mecanismo de consulta, almacenamiento, actualización y recuperación de objetos desde una base de datos.

Este lenguaje, que recibe el nombre de *Hibernate Query Lenguaje* (HQL), es una extensión orientada a objetos de SQL

El uso de HQL nos permite usar un lenguaje intermedio que según la base de datos que usemos y el dialecto que especifiquemos será traducido al SQL dependiente de cada base de datos de forma automática y transparente.

A modo de ejemplo se describen los pasos que sigue *Hibernate* para resolver una consulta sencilla como `FROM Persona P WHERE P.Nombre = 'Yankoa'`:

1. Determina el dialecto SQL que se esta utilizando a través del archivo de configuración de *Hibernate*.
2. Traduce la consulta al dialecto SQL que necesite.

```
SELECT Id, Nombre, Apellidos  
FROM Personas  
WHERE Persona.Nombre = 'Yankoa'
```

3. Ejecuta la consulta vía JDBC y obtiene un *ResultSet*.



4. Construye un objeto a partir de los elementos proporcionados en el ResultSet.

2.2.4.- JAVA

2.2.4.1.- INTRODUCCIÓN

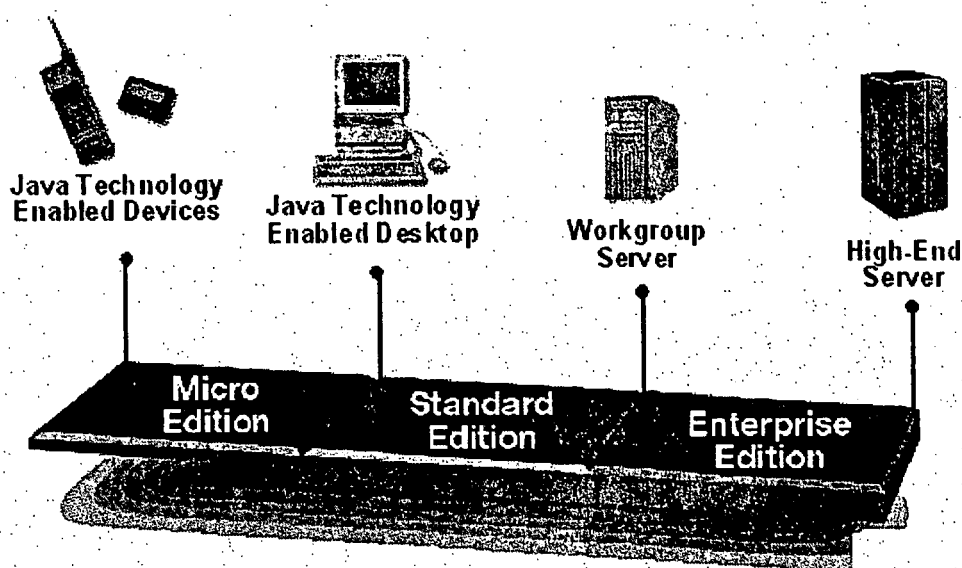
Los lenguajes estructurados se basan en estructuras de control de bloques de código y subrutinas independientes que soportan recursividad y variables locales. La programación orientada a objetos (POO) toma las mejores ideas de la programación estructurada y las combina con nuevos conceptos de organización, permitiendo descomponer un programa en grupos relacionados. Cada subgrupo pasa a ser un objeto autocontenido con sus propias instrucciones y datos.

Sun describe el lenguaje Java como “simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico”.

La existencia de distintos tipos de procesadores y ordenadores, llevó a la conclusión de que era muy importante conseguir un software que no dependiera del tipo de procesador utilizado. Se planteó la necesidad de conseguir código capaz de ejecutarse en cualquier tipo de máquina. Una vez compilado no debería ser necesaria ninguna modificación por el hecho de cambiar de procesador o de ejecutarlo en otra máquina. La clave consistió en desarrollar un código neutro el cual estuviera preparado para ser ejecutado sobre una máquina virtual denominada Java Virtual Machine (JVM).

La JVM es el intérprete de Java, es quien se encarga de ejecutar los ficheros con extensión .class creados por el compilador de Java (javac.exe)

Java ha desarrollado tres ediciones de plataformas diferentes: J2SE, J2EE y J2ME.



- Figura 9: Ediciones Java -

2.2.4.2.- JAVA 2 PLATFORM, STANDARD EDITION (J2SE)

Esta edición de Java es la más utilizada, y es la que recoge en cierta manera la iniciativa del lenguaje Java.

Contiene un conjunto de ejecución y de APIs para crear aplicaciones no destinadas a ser centralizadas y de accesos concurrentes.

2.2.4.3.- JAVA 2 PLATFORM, ENTERPRISE EDITION (J2EE)

Esta versión está orientada al entorno empresarial, ya que este entorno tiene unas características especiales: está pensado no para ser ejecutado en un equipo, sino para ejecutarse sobre una red de ordenadores de manera distribuida y remota.

Está especialmente enfocada al desarrollo de aplicaciones Web, ya que J2EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML ... y define cómo coordinarlos. J2EE también configura algunas especificaciones únicas para componentes. Estas incluyen Enterprise JavaBeans, servlets, JavaServer Pages y varias tecnologías de servicios Web. Esto permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores.

Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse



más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

2.2.4.4.- JAVA 2 PLATFORM, MICRO EDITION (J2ME)

Esta versión de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, PDAs o electrodomésticos inteligentes.

Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que sólo requiere unos pocos kilobytes de memoria para funcionar), la inclusión de un pequeño recolector de basura...

2.2.4.5.- COMENTARIOS

Para el desarrollo de este proyecto, como es evidente, se ha utilizado Java en su edición Enterprise Edition, ya que es la que mejor se adaptaba a las necesidades de un proyecto Web, en el que se necesita compartir información a través de la red.

2.2.5.- MYSQL

2.2.5.1.- INTRODUCCIÓN

MySQL, es un sistema de gestión de bases de datos SQL de código abierto (Open Source) desarrollado, distribuido y soportado por la compañía MySQL AB. MySQL AB es una compañía comercial, fundada por los desarrolladores de MySQL. Esta compañía Open Source de segunda generación une los valores y metodología Open Source con un exitoso modelo de negocio. El objetivo que persigue esta empresa consiste en que MySQL cumpla el estándar SQL, pero sin sacrificar velocidad, fiabilidad o usabilidad.

SQL (*Lenguaje de Consulta Estructurado*) fue comercializado por primera vez en 1981 por IBM, el cual fue presentado a ANSI y desde ese entonces ha sido considerado como un estándar para las bases de datos relacionales. Desde 1986, el estándar SQL ha aparecido en diferentes versiones como por ejemplo: SQL:92, SQL:99, SQL:2003.



2.2.5.2.- USO DE MYSQL

Con MySQL, podemos gestionar bases de datos normales y distribuidas.

Gracias a este sistema podemos crear diferentes tablas mediante las cuales se podrá almacenar datos manteniéndolos de esta forma centralizados. Además permite la creación de disparadores ó triggers, la realización de consultas, la creación de vistas, de cursores, de procedimientos almacenados...

La centralización de la información supone un gran avance tanto en lo referente al almacenamiento y distribución de datos y, tanto o más, en cuanto a medidas de seguridad para evitar la pérdida de información (de esto se encarga una parte muy importante de las tecnologías de la información, que es la Auditoría Informática).

2.2.5.3.- TIPOS DE DATOS

Después de la fase de diseño de una base de datos, y una vez se ha realizado el paso a tablas del mismo, es necesario crear las tablas correspondientes dentro de la base de datos. Para cada campo de cada una de las tablas, es necesario determinar el tipo de datos que contiene, para de esa forma ajustar el diseño de la base de datos, y conseguir un almacenamiento óptimo con la menor utilización de espacio.

Los tipos de datos que pueden existir en un campo, se pueden agrupar en tres grandes grupos:

- **Tipos numéricos**

Tipos de Campo	Tamaño de Almacenamiento
<i>tinyint</i>	1 byte
<i>smallint</i>	2 bytes
<i>mediumint</i>	3 bytes
<i>int</i>	4 bytes
<i>integer</i>	4 bytes
<i>bigint</i>	8 bytes
<i>float(x)</i>	4 ó 8 bytes
<i>float</i>	4 bytes
<i>double</i>	8 bytes
<i>double precision</i>	8 bytes
<i>real</i>	8 bytes
<i>decimal(M,D)</i>	M+2 bytes si $D > 0$, M+1 bytes si $D = 0$
<i>numeric(M,D)</i>	M+2 bytes if $D > 0$, M+1 bytes if $D = 0$

- Tabla 16: MySQL datos numéricos -



- Tipos de fecha

Tipo de Campo	Tamaño de Almacenamiento
<i>date</i>	3 bytes
<i>datetime</i>	8 bytes
<i>timestamp</i>	4 bytes
<i>time</i>	3 bytes
<i>year</i>	1 byte

- Tabla 17: MySQL datos tipo fecha -

- Tipos de cadena

Tipo de campo	Tamaño de Almacenamiento
<i>char(n)</i>	n bytes
<i>varchar(n)</i>	n +1 bytes
<i>tinyblob, tinytext</i>	Longitud+1 bytes
<i>blob, text</i>	Longitud +2 bytes
<i>mediumblob, mediumtext</i>	Longitud +3 bytes
<i>longblob, longtext</i>	Longitud +4 bytes
<i>enum('value1','value2',...)</i>	1 ó dos bytes dependiendo del número de valores
<i>set('value1','value2',...)</i>	1, 2, 3, 4 ó 8 bytes, dependiendo del número de valores

- Tabla 18: MySQL datos tipo cadena -

2.2.5.4.- COMENTARIOS

MySQL es muy utilizado en aplicaciones Web y es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones Web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones y por lo tanto para este Proyecto Fin de Carrera.



2.2.6.- APACHE TOMCAT

2.2.6.1.- INTRODUCCIÓN

Tomcat es un contenedor de Servlet, que implementa las especificaciones de Servlet y Java Server Pages (JSP). Fue desarrollado bajo el proyecto de *Jakarta* en la Apache Software Foundation.

En sus comienzos existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor Web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java.

2.2.6.2.- FICHEROS DE CONFIGURACIÓN

La configuración de Tomcat se basa en dos ficheros:

- **Server.xml**

Es el fichero de configuración principal de Tomcat y tiene dos objetivos. Por un lado se encarga de proporcionar la configuración inicial para los componentes de Tomcat y por otro lado se encarga de especificar su estructura, es decir, permite que Tomcat arranque y se construya por si mismo.

Los elementos más importantes de server.xml se describen en la siguiente tabla:



Elemento	Descripción
Server	Define un servidor Tomcat. Puede contener elementos Logger y ContextManager.
Logger	Define un objeto logger. Cada objeto de este tipo tiene un nombre que lo identifica, así como un path para el fichero log que contiene la salida.
ContextManager	Especifica la configuración y estructura para un conjunto de ContextInterceptor, Context y sus componentes.
ContextInterceptor y RequestInterceptor	Estos interceptores, escuchan ciertos eventos que suceden en el ContextManager, como por ejemplo, los eventos de arranque y parada de Tomcat.
Connector	Representa una conexión al usuario, a través de un servidor Web o directamente al navegador. Este objeto es responsable del control de threads y de leer/escribir la peticiones/respuestas desde los sockets conectados a diferentes clientes.
Context	Cada context representa un path en el árbol de Tomcat donde situar nuestra aplicación Web. Contiene los siguientes elementos: <ul style="list-style-type: none">- El path donde se localiza el contexto. Puede ser completo ó relativo al home del ContextManager.- Nivel de depuración usado para los mensajes de depuración.- Una bandera reloadable que permite recargar Tomcat cuando hay un cambio de forma que se pruebe el nuevo código sin tener que parar y arrancar el servidor.

- Tabla 19: Elementos del fichero Server.xml -

2.2.6.3.- COMENTARIOS

Todos estos elementos, y entre otros motivos su sencillez, tanto en el uso como en la configuración, hacen que Tomcat fuera elegido como servidor Web para esta aplicación.

2.2.7.- JAVA SERVER PAGES (JSP)

Las páginas JSP, desarrolladas por Sun Microsystems, son un tipo especial de páginas que combinan HTML con fragmentos de Java para producir páginas Web dinámicas.

Su funcionamiento es el siguiente:



Cuando un cliente pide una página JSP del sitio Web y esta no se ha ejecutado antes, la página es inicialmente pasada al motor JSP, el cual se encarga de compilarla y convertirla en Servlet. Después la ejecuta y devuelve el contenido resultante al cliente en forma de página HTML.

La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java lo que permite separar en niveles o capas las aplicaciones Web, almacenando en clases Java las partes que consumen más recursos (así como las que requieren más seguridad) y dejando la parte encargada de formatear el documento HTML en el archivo JSP. La idea fundamental detrás de este criterio es el de separar la lógica del negocio de la presentación de la información.

2.2.8.- JAVA SERVER PAGES STANDARD TAG LIBRARY (JSTL)

La librería JSTL es un componente dentro de la especificación de J2EE y es controlada por Sun Microsystems.

JSTL es un conjunto de librerías de etiquetas simples y estándares que permiten extender funcionalidades personalizadas por medio de TAGs incluidos dentro de las páginas JSP, invocando una determinada funcionalidad sin tener que escribir código Java en la página.

Las etiquetas JSTL se encuentran organizadas en 4 librerías:

- **Etiquetas Core**

Etiquetas para iteración sobre datos, operaciones condicionales, e importación de datos de otras páginas.

Etiqueta	Descripción
c:out	Muestra el resultado de una expresión. Su funcionalidad es equivalente a la de <code><%= %></code> .
c:set	Introduce/Modifica un valor en una variable.
c:remove	Elimina una variable.
c:if	Evalúa si una condición se cumple.
c:choose, c:when, c:otherwise	Evalúa si una condición se cumple y sino se cumple.
c:forEach	Permite iterar sobre los elementos siguientes: <ul style="list-style-type: none">- Arrays de objetos o tipos primitivos.- Instancias de <code>java.util.Collection</code>, <code>java.util.Map</code>, <code>java.util.Iterator</code>, <code>java.util.Enumeration</code>.- Cadenas delimitadas por comas.- Instancias de <code>javax.servlet.jsp.jstl.sql.Result</code>.
c:forEachTokens	Permite descomponer una cadena en tokens.
c:import	Proporciona toda la funcionalidad de <code>jsp:include</code> .



c:param	c:import puede tener hijos c:param, que se usan para pasar parámetros a la URL a recuperar.
c:redirect	Ejecuta un forward.
c:url	Sirve para mostrar una URL o para grabarla en una variable.
c:catch	Permite capturar y tratar los errores que se produzcan en una página.

- Tabla 20: Etiquetas Core -

- **Etiquetas Xml**

La librería XML de JSTL usa XPath para obtener partes de los documentos XML con que opera, es decir, comprende el procesamiento de XML.

En JSTL, cuando una expresión XPath devuelve algún nodo, se considera true, si no, false.

Una variable XPath tiene el formato \$nombre, o el formato \$nombre1:nombre2. Por ejemplo: \$sessionScope:usuario.

- **Etiquetas Fmt**

Comprende la internacionalización y formato de valores como de moneda y fechas.

Etiqueta	Descripción
fmt:bundle	Permite cargar recursos de mensajes específicos solo para una zona de la página
fmt:message	Acepta una clave y devuelve el resultado correspondiente a la clave en el fichero de recursos especificado.
fmt:param	Algunos mensajes aceptan parámetros, de forma parecida a los PreparedStatement de SQL.
fmt:setBundle	Permite cargar recursos de mensajes específicos.
fmt:setLocale	Permite establecer el valor de la Locale.
fmt:formatNumber	Permite dar formato a un valor numérico.
fmt:formatDate	Permite dar formato a una fecha.
fmt:parseNumber	Interpreta cadenas como números. Es necesario para pasar valores numéricos a algunas etiquetas como sql:param, fmt:formatNumber, y otras.
fmt:parseDate	Interpreta cadenas como fechas. Usa el locale del contenedor JSP.
fmt:setTimeZone	Permite cambiar la zona horaria.
fmt:timeZone	Permite cambiar temporalmente la zona horaria.

- Tabla 21: Etiquetas Fmt -



- **Etiquetas Sql**

Comprende el acceso a base de datos.

2.2.9. - JAVASCRIPT

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores Web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

Este lenguaje consta de tres partes principales:

- **Client-side JavaScript:** La parte cliente de JavaScript permite la creación de aplicaciones que se ejecutan en un navegador.
- **Server-side JavaScript:** La parte servidor de JavaScript permite la creación de aplicaciones que se ejecutan en un servidor Web.
- **Core JavaScript:** La parte central de JavaScript contiene un conjunto de objetos básicos (Array, Date, Math...) además de los elementos del lenguaje como son los operadores, las estructuras de control y las sentencias.

Utilizando JavaScript se pueden crear páginas HTML de forma dinámica que procesan las entradas del usuario y mantienen datos persistentes utilizando objetos especiales, ficheros y bases de datos relacionales. Además ofrece la posibilidad de comunicarse con aplicaciones escritas en Java, de modo que se puedan instanciar objetos Java desde el código JavaScript y también se pueda acceder a los objetos, propiedades y métodos JavaScript desde código Java. También permite la validación de datos sin necesidad de acceso a un servidor.



2.2.10. - ECLIPSE WEB TOOLS PLATFORM (WTP)

2.2.10.1. - INTRODUCCIÓN

Eclipse es una plataforma de software de código abierto para desarrollar en Java, creada originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La definición que da el proyecto Eclipse acerca de su software es: *"una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular"*.

Esta plataforma es considerada un Entorno de Desarrollo Integrado (IDE), ya que con ella es posible escribir código Java, compilarlo y ejecutarlo sin tener que cambiar de aplicación.

Eclipse emplea módulos (plug-in) para proporcionar toda su funcionalidad al frente de la plataforma, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software.

En cuanto a las aplicaciones clientes, Eclipse provee al programador con marcos de trabajo (Frameworks) muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones Web, etc.

2.2.10.2.- CARACTERÍSTICAS

La versión actual de Eclipse dispone de las siguientes características:

- Editor de texto.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit, que es un conjunto de librerías creadas por Erich Gamma y Kent Beck utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.
- Control de versiones con Concurrent Versions System (CVS), que es una aplicación informática que mantiene el registro de todo el trabajo y los cambios en los ficheros (código fuente principalmente) que forman un proyecto permitiendo que distintos desarrolladores (potencialmente situados a gran distancia) colaboren.
- Integración con Apache Ant, que es una herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción.
- Asistentes (wizards): para creación de proyectos, clases, tests, etc.



- Refactorización, que es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo.

2.2.10.3.- WTP

WTP es un plugin de Eclipse que proporciona APIs específicas para J2EE centrados en el desarrollo de aplicaciones Web.

Los principios clave sobre los que el proyecto Eclipse WTP funciona, son los siguientes:

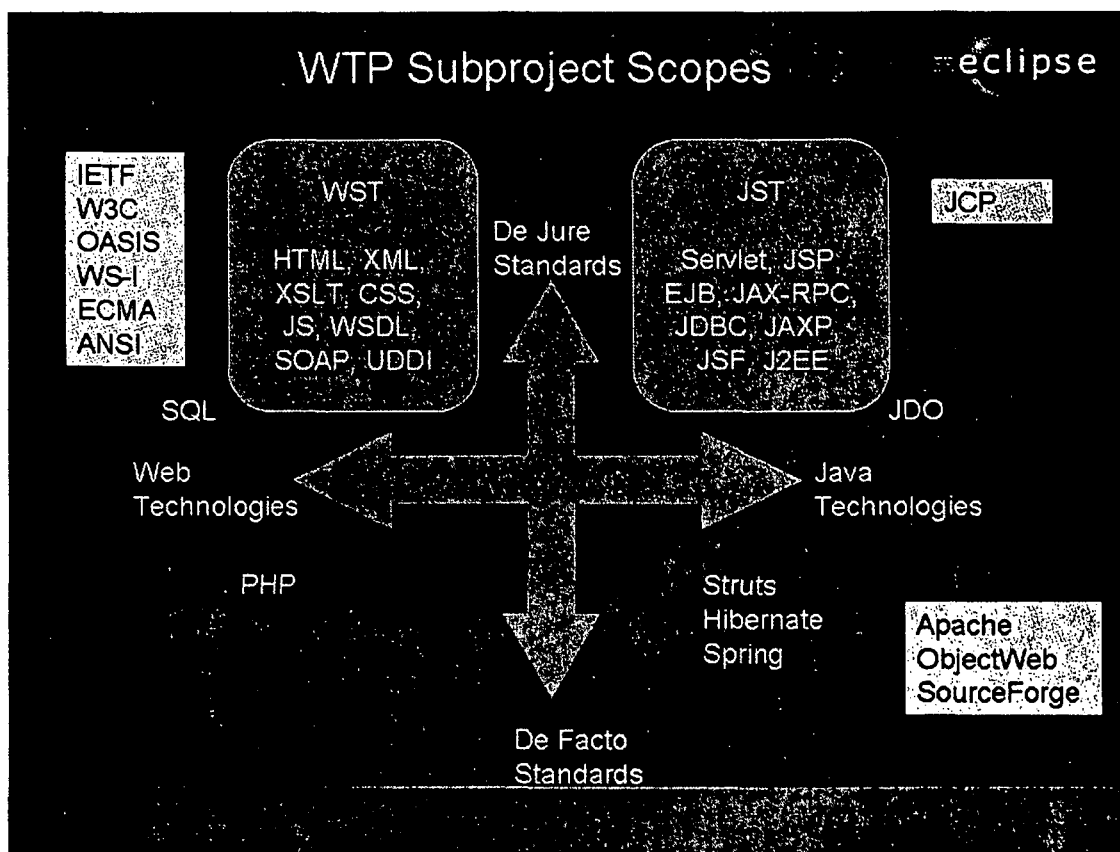
- **Extensión del valor de propuesta de Eclipse** El Proyecto Eclipse ha establecido un alto estándar de excelencia técnica, innovación funcional y una extensibilidad general dentro del dominio de Java IDE. Intenta aplicar estos mismos estándares al dominio de herramientas Web/J2EE.
- **Ecosistema del vendedor** El mayor objetivo de este proyecto es dar soporte a una herramienta vital de desarrollo de aplicaciones del mercado. Su base ejemplar será útil por sí misma pero será diseñada en principio para ser extensible; así los vendedores comerciales pueden usar los resultados generados por este proyecto para su propia innovación de producto y eficiencia de desarrollo.
- **Neutralidad del vendedor** La neutralidad del vendedor estará situada en el núcleo de este proyecto. Se intenta motivar la participación en Eclipse y conducir su aceptación en el mercado mediante el fortalecimiento de las valiosas propuestas, a largo plazo del producto, del más amplio sector de vendedores de desarrollo de aplicaciones.
- **Innovación basada en estándares** Este proyecto generará una base de herramientas basadas en estándares extensibles en el cual un amplio sector de vendedores podrá crear valiosos productos de desarrollo para sus clientes y usuarios. Donde quiera que exista un estándar, nos ceñiremos a él. Al menos, en principio, donde emerjan estándares, esperaremos a que emerjan; esto puede ser reevaluado más tarde de acuerdo a las necesidades del usuario y disponibilidad del colaborador.
- **Desarrollo ágil** El objetivo es incorporar en el proceso de planificación las innovaciones que aparezcan una vez que un proyecto está en marcha y conseguir el apoyo de la comunidad de usuarios en lo que se logre hasta la fecha. Se piensa en un proceso de desarrollo y planificación ágil, en el cual en progreso sea incremental.
- **Diversidad y gama** Pretenden asimilar las mejores ideas del gran número



de participantes que representan las necesidades del enorme sector de usuarios finales. Por eso animan a las organizaciones a lo largo del sector técnico, mercado y dominios geográficos a participar en este proyecto.

En cuanto al alcance de este proyecto, al ser tan amplio, se podría decir que abarca todos los aspectos del desarrollo de software, pero realmente se concentra inicialmente en la infraestructura para herramientas utilizadas en la construcción de aplicaciones para entornos de ejecución Web y Java basados en estándares.

Fuera del alcance del proyecto, al menos en principio, existirá un soporte específico para arquitecturas de aplicación específicas para el vendedor, como ASP.Net y ColdFusion, o para extensiones no respaldadas por el JCP, como Apache Struts. Esto puede ser reevaluado más adelante de acuerdo a las necesidades del usuario y disponibilidad del colaborador y podría conducir a la creación de un nuevo subproyecto.



- Figura 10: Alcance de los subproyectos de WTP -

El proyecto esta limitado a proveer una infraestructura para el uso de herramientas en sí, en contraste con la infraestructura referente a la aplicación en tiempo de ejecución.



El objetivo final del proyecto es dar soporte a el uso de herramientas que permitan a los desarrolladores producir aplicaciones que otorguen un amplio grado de extensibilidad y reutilización junto con una eficiencia de desarrollo creciente.

2.2.10.4.- COMENTARIOS

Para la realización de este Proyecto Fin de Carrera se ha optado por usar Eclipse WTP, ya que contiene ya integrados todos los elementos necesarios para crear una aplicación Web partiendo desde cero.

2.2.11.- SQLYOG

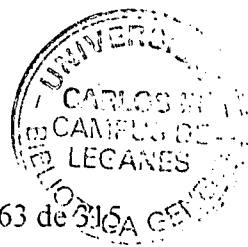
SQLyog es una interfaz gráfica diseñada para trabajar de forma más rápida y cómoda con el servidor de base de datos MySQL.

El programa está orientado a usuarios que ya posean ciertos conocimientos de SQL y necesiten un intérprete gráfico sin excesivas florituras, ágil y funcional. SQLyog permite administrar usuarios y permisos, y realizar múltiples peticiones a base de datos.

También puedes insertar fácilmente, mediante plantillas, peticiones que se realicen con frecuencia, como por ejemplo creación de tablas. Otras opciones son la exportación de datos en formato CSV, HTML y XML, la optimización de bases de datos, la posibilidad de guardar scripts de SQL como favoritos y más.

2.2.12.- OTRAS HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

- Dreamweaver.
- Adobe PhotoShop.
- Strut Console.
- Hibernate Tools.
- Software Reuse Studio.
- Smart Draw 7.
- Microsoft Office XP.
- Windows XP.
- Internet Explorer 6.
- Microsoft Project 98.





3.- GESTIÓN DEL PROYECTO

3.1.- CICLO DE VIDA

Se ha seleccionado el ciclo de vida en cascada ya que da un enfoque sistemático y secuencial del desarrollo del software hasta conseguir el producto final.

Las principales actividades del ciclo de vida en cascada son:

Análisis de los requisitos y planificación: En ella se recolectan los requisitos, alternativas y restricciones, además se estima y planifica el proyecto.

Revisión del análisis de los requisitos y planificación: Se revisan los puntos más importantes de las tareas realizadas en la actividad anterior, se generan los informes y se evalúan los resultados. Además se llevan a cabo reuniones con el cliente para validar los requisitos o añadir más conforme a lo que se tiene.

Diseño: Consiste en traducir los requisitos en una representación del software que pueda ser establecida de forma que se obtenga la calidad requerida.

Revisión del diseño: Se revisan los puntos más importantes de las tareas realizadas en actividad anterior, se generan los informes y se evalúan los resultados. Además se llevan a cabo reuniones con el cliente para conocer si está conforme con el diseño generado en los mismos y validarlo.

Codificación: Utilizando las herramientas software seleccionadas en la fase de análisis se procede a la implementación del proyecto software de manera que la máquina sea capaz de interpretarlo.

Revisión de codificación: Se revisa la codificación del proyecto y refina para reparar errores. Tras una reunión se valida la codificación.

Pruebas: En esta fase se realizan las pruebas y revisión del código comprobando que se obtienen los resultados requeridos.

Mantenimiento: Puesto que el software sufrirá cambios tras la entrega al cliente en esta etapa se adapta el software a los requerimientos del cliente y de los avances tecnológicos.

Se especifica que el uso de este ciclo de vida se ha elegido dado que una de sus características es que desde cualquier tarea se puede volver a cualquiera de las anteriores en caso que sea necesario.



3.2.- ORGANIZACIÓN DEL PROYECTO

3.2.1.- WORK BREAKDOWN STRUCTURE (WBS)

3.2.1.1.- DESCRIPCIÓN DE LAS TAREAS Y ACTIVIDADES DEL ESTÁNDAR IEEE1074

1. Procesos de gestión

1.1 Iniciación de actividades del proyecto

1.1.1 Creación del ciclo de vida: En base a la realización de actividades de control, implementación de documentos y a la fabricación de los materiales de pruebas se elegirá el ciclo de vida del proyecto software.

1.1.2 Realización de estimaciones: A partir de los requisitos documentados, se evaluarán el tamaño, los costes y el esfuerzo para realizar las actividades del ciclo de vida. También se deben estudiar los recursos de la computadora donde se vaya a instalar el proyecto.

1.1.3 Distribución de los recursos del proyecto: Se deben identificar las actividades dentro del ciclo de vida y los recursos que contendrá el proyecto como son personal, equipo, tiempo, espacio...

1.1.4 Definición de métricas: Como su nombre indica, se definen todas las métricas a utilizar en el proyecto (IEEE1074, IEEE1058...).

1.2 Planificación de actividades del proyecto

1.2.1 Plan de evaluaciones: Consiste en evaluar si el desarrollo y la producción del software alcanzará las metas o el objetivo deseado. Para evaluarlos se necesitan realizar revisiones, auditorias y tests de pruebas. Con esto se evitarán los riesgos técnicos.

1.2.2 Plan de configuración del desarrollo: Usando unas líneas de seguimiento internas y preacordadas se realizaran documentos y un plan específico para la organización y gestión de herramientas, metodologías, personal, técnicas....

1.2.3 Plan de transición del sistema: Consiste en la sustitución de un sistema antiguo u obsoleto por uno nuevo. Se requiere documentación acerca de las herramientas, metodologías, personal... implicado en dicha transición.

1.2.4 Plan de instalación: Se analizan los conocimientos del instalador y los requisitos de la instalación para crear una documentación adecuada para realizarla correctamente.

1.2.5 Plan de documentación: Se crea un documento en el que aparece detallada toda la información de la que se dispone en ese momento.

1.2.6 Pruebas: Se deben identificar las necesidades del personal de pruebas para realizarlas sobre el proyecto. Estas necesidades deben recoger tanto desde un punto de vista interno (p.e, equipo de proyecto)



como las externas (p.e, cliente, usuario). Deben ser revisados los requisitos del cliente junto con los inventarios personales.

1.2.7 Desarrollo del plan del proyecto: Detalla la asignación del proyecto y las responsabilidades. Estándares, metodologías y herramientas para la gestión de configuración, calidad, evaluación, ensayos, documentación y desarrollo.

1.2.8 Plan de integración: Se analiza el diseño y requisitos del software para determinar el orden de combinación de los componentes del software en general. Hay que documentar los métodos de integración.

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Se analizan y mitigan negocios, técnicas, economía, seguridad y riesgos de seguridad.

1.3.2 Gestión del proyecto: Se revisa el progreso del proyecto y se valora de acuerdo a los planes y estimaciones establecidas anteriormente.

1.3.3 Identificar las mejoras necesarias del ciclo de vida: Se evalúan los informes generados para valorar que cambios podrían ser beneficiosos. Se emplea para esto diagramas de control, procesos de valoración de capacidad...

1.3.4 Conservación de registros: Recoge registros del proyecto de distintos grupos de actividad. Estos formarán parte de un histórico e incluyen planificaciones futuras, contabilidad corporativa...

1.3.5 Recogida y análisis de datos de la métrica: Analiza los datos generados por la métrica empleada, informes de evaluación, información de entrada de clientes...

2. Grupos de actividad previos al desarrollo

2.1 Actividades de exploración conceptual

2.1.1 Identificación de ideas o necesidades: Se generan ideas o necesidades para un sistema modificado o nuevo.

2.1.2 Formulación de métodos potenciales: De acuerdo al presupuesto y los informes generados, se genera un documento preliminar en el que se describen la decisión de adquirir el sistema o no.

2.1.3 Estudio de las conductas de viabilidad: Debe recoger el análisis de una idea o necesidad, los métodos potenciales y las condiciones y beneficios del ciclo de vida empleado. Hay que decidir además si se crea o se compra el sistema, en parte o en su totalidad.

2.1.4 Refinar y finalizar la idea o necesidad: Consiste en analizar las necesidades establecidas con anterioridad, los métodos potenciales las recomendaciones y el impacto de la transición de un sistema a otro, si procede. El método se debe seleccionar y documentar de manera que sea capaz de refinar la idea o necesidades iniciales.



2.2 Sistema de asignación de actividades

2.2.1 Analizar funciones: Se analizan la totalidad de las funciones. Una vez identificadas se emplean para desarrollar la arquitectura del sistema e identificar las funciones software y hardware.

2.2.2 Desarrollo de la arquitectura del sistema: Resulta de la transformación de la descripción funcional del sistema mediante metodologías estándares y herramientas establecidas por la empresa. Se convierte así en la base del diseño de la actividad de grupo para determinar las funciones del software y hardware.

2.2.3 Descomponer requisitos del sistema: Las funciones del sistema documentadas anteriormente se dividen, acorde al sistema de arquitectura, para formar los requisitos software humanos y los requisitos hardware. Este sistema distingue las interfaces externas al sistema de los objetos que conforman el sistema.

2.3 Actividades importadas de software

2.3.1 Identificar requisitos importados del sistema: Se extrae aquellos requisitos de software mejor satisfechos por software existente o adquirido. El resultado cubrirá todas las categorías de requisitos, incluyendo la planificación y restricciones de presupuesto.

2.3.2 Evaluar los recursos importados de software: Se trata de un mecanismo que determina si los requisitos importados de software son satisfechos usando el software de otro proyecto.

2.3.3 Definir los métodos de software importados: Esta actividad selecciona los métodos más apropiados con los cuales los recursos importados de software seleccionados proveerán el software importado.

2.3.4 Importar software: Consiste en importar software para usarlo en el proyecto. Se realiza la importación de tal manera que se asegure la integridad de la totalidad del sistema.

3. Desarrollo de actividades de grupo

3.1 Actividades requeridas

3.1.1 Definir y desarrollar requisitos software: Se analiza el software funcional para determinar la claridad, validez, seguridad. Es necesario el uso de una metodología comprensiva para asegurar que los requisitos son completos y consistentes. Se emplean técnicas de análisis estructurado, modelado, prototipado...

3.1.2 Definir requisitos de interfaz: Debe especificar todos los interfaces software requeridos para mantener el desarrollo y ejecución del software del sistema.

3.1.3 Priorizar e integrar requisitos software: La funcionalidad y rendimiento de los requisitos deben ser revisados y repasados. La



organización de los requisitos de software a su vez debe ser revisada todo lo que sea necesario.

3.2 Diseño de actividades

3.2.1 Ejecutar el diseño de arquitectura: Transforma los requisitos del software y el sistema arquitectónico a un nivel conceptual alto. Durante esta actividad los componentes software que constituyen el software del sistema y sus estructuras son identificados. El software adquirido y los contenidos de las librerías software pueden influir en el diseño de la arquitectura.

3.2.2 Diseñar la base de datos: Esta actividad debe especificar la estructura de información que esta esquematizada en los requisitos software y sus características del sistema software.

3.2.3 Diseñar interfaces: Deben concernir al usuario, software e interfaces del hardware del sistema contenidos en los requisitos del sistema. Esta actividad debería consolidar estas descripciones de interfaz en un único diseño de interfaz para el software del sistema.

3.2.4 Ejecutar los diseños detallados: En esta actividad las alternativas de diseño deberán ser elegidas para implementar las funciones especificadas por cada componente software. Al final de esta actividad la estructura de datos, algoritmos e información de control de cada componente software es especificada.

3.3 Implementación de actividades

3.3.1 Crear código ejecutable: Debe ser generado tomando como referencia el ciclo de vida del software.

Este código fuente debe ser agrupado en unidades procesables, lo cual estará determinado por el lenguaje de programación seleccionado y el diseño de la información. Además todas las actividades se transformarán a código ejecutable y depurado.

3.3.2 Crear la documentación operativa: Se debe producir la documentación del proyecto en cuanto a detalles de diseño y requisitos de la interfaz software. Esto es necesario para la instalación y soporte del sistema al margen del ciclo de vida.

3.3.3 Integración de la ejecución: Implementa la base de datos, el código fuente, el código ejecutable y la información de documentación planificada en el software.

4. Post desarrollo de los grupos de actividades

4.1 Instalación de actividades

4.1.1 Distribuir el software: El testeo, la base de datos, la documentación de operación y la información del plan de instalación del software, se empaquetarán en sus medios de almacenamiento y se



distribuye a aquellos lugares donde se pretenda instalar.

4.1.2 Instalar el software: Se lleva a cabo la instalación del software en el entorno de trabajo deseado.

4.1.3 Aceptar el software en el entorno operacional: Se realiza un análisis del informe producido de acuerdo con el plan de conformidad del usuario, para comprobar que el software instalado funciona correctamente.

4.2 Operación y soporte de actividades

4.2.1 Manejar el sistema: El software de sistema instalado, se utilizará en el entorno y de acuerdo con las instrucciones operativas.

4.2.2 Proporcionar asistencia técnica y consultoría: Se aplica después de ser aceptada por el usuario. El soporte debe contestar preguntas técnicas del usuario.

4.2.3 Mantener el soporte para el Log de peticiones: Almacena las peticiones en el Log de soporte de peticiones.

4.3 Actividades de mantenimiento

4.3.1 Identificar las necesidades del software: Se identifican las lecciones aprendidas y necesarias, para mejorar el software.

4.3.2 Implementar un método de aviso de problemas: Se aceptan anomalías de cualquier recurso y genera un mensaje de error.

4.3.3 Replicar el ciclo de vida del software: Monitoriza problemas en los esfuerzos de corrección que se realizan por el responsable del grupo de actividad. Debe comprobar que la implementación de la solución por el responsable del grupo de actividad ha sido comprobada y debe almacenar la resolución del problema en un informe de resolución de problemas.

4.4 Retirada de actividades

4.4.1 Aviso al usuario: Se avisa a todos los usuarios formalmente de que se retira un sistema de la actividad.

4.4.2 Operaciones en paralelo: Se debe aplicar solo si el sistema se va a sustituir por uno nuevo, por lo tanto, esta actividad requiere un periodo de retirada del sistema actual mientras se prepara la entrada del nuevo sistema.

4.4.3 Retirada del sistema: Consiste en la retirada y salvaguarda de la información del sistema actual. El usuario debe ser notificado del cambio.



5. Actividades integrales

5.1 Evaluación de actividades

5.1.1 Revisión de conductas: Revisión de todas las actividades del ciclo de vida.

5.1.2 Crear una matriz de rastreo: Debe mostrar los requisitos y recursos utilizados a lo largo del ciclo de vida del proyecto.

5.1.3 Auditorias de conducta: Examina independientemente los procesos o productos software.

5.1.4 Desarrollo de procesos de prueba: Realiza pruebas en cada uno de los niveles, indicando qué tipo de prueba se realiza, qué se está probando, qué datos se están usando... y los resultados esperados.

5.1.5 Crear pruebas de datos: Se ejecutan las pruebas de datos y se evalúan.

5.1.6 Ejecutar pruebas: Se ejecutan las pruebas pertinentes y se comparan los resultados obtenidos con los esperados. En caso de anomalía, se genera un informe con dichas anomalías.

5.1.7 Notificar los resultados de evaluación: Reúne la información y los datos proporcionados por la información de entrada. Se generan resultados y se notifican.

5.2 Actividades de gestión de configuración de software

5.2.1 Desarrollar identificaciones de configuración: Se definen los identificadores de configuración incluyendo las líneas base, títulos...

5.2.2 Realizar control de configuraciones: Controla la configuración del producto.

5.2.3 Realizar recuentos de estados: Recibe la identificación de configuración y cambio de estado y debe crear una actualización de la información del estado.

5.3 Documentación de las actividades de desarrollo

5.3.1 Implementar la documentación: Incluye el diseño, preparación y mantenimiento de la documentación.

5.3.2 Producir y distribuir la documentación: Se recolecta la información en un único documento.

5.4 Actividades de entrenamiento

5.4.1 Desarrollar los materiales de entrenamiento: Se identifican y se revisan todos los materiales y la información de entrada que sea necesaria para el objetivo.

5.4.2 Validar el programa de entrenamiento: Evalúa la efectividad de la entrega y validación del material presentado.

5.4.3 Implementar el programa de entrenamiento: Se asegura de que existan provisiones de todos los materiales del acceso a las



localizaciones y facilidades para el entrenamiento, además de la entrega del entrenamiento.

3.2.1.2.- ESTRUCTURACIÓN DE TAREAS

ANÁLISIS DE REQUISITOS Y PLANIFICACIÓN

- 1.1 Iniciación de actividades del proyecto
 - 1.1.1 Creación del ciclo de vida
 - 1.1.2 Realización de estimaciones
 - 1.1.3 Distribución de los recursos del proyecto
 - 1.1.4 Definición de métricas
- 1.2 Planificación de actividades del proyecto
 - 1.2.2 Plan de configuración del desarrollo
 - 1.2.4 Plan de instalación
 - 1.2.5 Plan de documentación
 - 1.2.6 Pruebas
 - 1.2.7 Desarrollo del plan del proyecto
- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
- 2.1 Actividades de exploración conceptual
 - 2.1.1 Identificación de ideas o necesidades
- 3.1 Actividades requeridas
 - 3.1.1 Definir y desarrollar requisitos software
- 5.1 Evaluación de actividades
 - 5.1.2 Crear una matriz de rastreo
 - 5.1.7 Notificar los resultados de evaluación
- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados

REVISIÓN DEL ANÁLISIS DE REQUISITOS Y PLANIFICACIÓN

- 1.2 Planificación de actividades del proyecto
 - 1.2.5 Plan de documentación
 - 1.2.7 Desarrollo del plan del proyecto
- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
 - 1.3.3 Identificar las mejoras necesarias del ciclo de vida
 - 1.3.5 Recogida y análisis de datos de la métrica
- 3.1 Actividades requeridas
 - 3.1.1 Definir y desarrollar requisitos software
- 5.1 Evaluación de actividades



- 5.1.4 Desarrollo de procesos de prueba
- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados

DISEÑO

- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
- 2.2 Sistema de asignación de actividades
 - 2.2.1 Analizar funciones
 - 2.2.2 Desarrollo de la arquitectura del sistema
 - 2.2.3 Descomponer requisitos del sistema
- 3.1 Actividades requeridas
 - 3.1.1 Definir y desarrollar requisitos software
 - 3.1.2 Definir requisitos de interfaz
- 3.2 Diseño de actividades
 - 3.2.1 Ejecutar el diseño de arquitectura
 - 3.2.2 Diseñar la base de datos
 - 3.2.3 Diseñar interfaces
- 5.2 Actividades de gestión de configuración de software
 - 5.2.1 Desarrollar identificaciones de configuración
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados
- 5.3 Documentación de las actividades de desarrollo
 - 5.3.1 Implementar la documentación

REVISIÓN DEL DISEÑO

- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
 - 1.3.3 Identificar las mejoras necesarias del ciclo de vida
- 2.2 Sistema de asignación de actividades
 - 2.2.2 Desarrollo de la arquitectura del sistema
- 3.1 Actividades requeridas
 - 3.1.1 Definir y desarrollar requisitos software:
 - 3.1.3 Priorizar e integrar requisitos software
- 3.2 Diseño de actividades
 - 3.2.2 Diseñar la base de datos
- 5.1 Evaluación de actividades
 - 5.1.4 Desarrollo de procesos de prueba:
- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados



CODIFICACIÓN

- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
- 3.2 Diseño de actividades
 - 3.2.4 Ejecutar los diseños detallados
- 3.3 Implementación de actividades
 - 3.3.1 Crear código ejecutable
 - 3.3.3 Integración de la ejecución
- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados

REVISIÓN DE CODIFICACIÓN

- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
- 3.3 Implementación de actividades
 - 3.3.1 Crear código ejecutable
 - 3.3.2 Crear la documentación operativa
 - 3.3.3 Integración de la ejecución
- 5.1 Evaluación de actividades
 - 5.1.4 Desarrollo de procesos de prueba
- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados
- 5.3 Documentación de las actividades de desarrollo
 - 5.3.1 Implementar la documentación

PRUEBAS

- 1.2 Planificación de actividades del proyecto
 - 1.2.1 Plan de evaluaciones
 - 1.2.6 Pruebas
- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
- 4.1 Instalación de actividades
 - 4.1.1 Distribuir el software
 - 4.1.3 Aceptar el software en el entorno operacional
- 5.1 Evaluación de actividades
 - 5.1.4 Desarrollo de procesos de prueba
 - 5.1.5 Crear pruebas de datos
 - 5.1.6 Ejecutar pruebas



- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados

MANTENIMIENTO

- 1.3 Visualización y control de las actividades del proyecto
 - 1.3.1 Gestión de riesgos
 - 1.3.2 Gestión del proyecto
- 4.1 Instalación de actividades
 - 4.1.2 Instalar el software
- 4.2 Operación y soporte de actividades
 - 4.2.2 Proporcionar asistencia técnica y consultoría
- 5.2 Actividades de gestión de configuración de software
 - 5.2.2 Realizar control de configuraciones
 - 5.2.3 Realizar recuentos de estados
- 5.3 Documentación de las actividades de desarrollo
 - 5.3.2 Producir y distribuir la documentación



	Análisis de requisitos y planificación	Revisión del análisis	Diseño	Revisión del diseño	Codificación	Revisión de codificación	Pruebas	Mantenimiento
A.1.1.1	X							
A.1.1.2	X							
A.1.1.3	X							
A.1.1.4	X							
A.1.2.1							X	
A.1.2.2	X							
A.1.2.3								
A.1.2.4	X							
A.1.2.5	X	X						
A.1.2.6	X						X	
A.1.2.7	X	X						
A.1.2.8								
A.1.3.1	X	X	X	X	X	X	X	X
A.1.3.2	X	X	X	X	X	X	X	X
A.1.3.3		X		X		X		
A.1.3.4								
A.1.3.5		X						
A.2.1.1	X							
A.2.1.2								
A.2.1.3								
A.2.1.4								
A.2.2.1			X					
A.2.2.2			X	X				
A.2.2.3			X					
A.2.3.1								
A.2.3.2								
A.2.3.3								
A.2.3.4								
A.3.1.1	X	X	X	X				
A.3.1.2			X					
A.3.1.3				X				
A.3.2.1			X					
A.3.2.2			X	X				
A.3.2.3			X					
A.3.2.4					X			
A.3.3.1					X	X		
A.3.3.2						X		
A.3.3.3					X	X		
A.4.1.1							X	
A.4.1.2								X
A.4.1.3							X	
A.4.2.1								
A.4.2.2								X
A.4.2.3								
A.4.3.1								
A.4.3.2								
A.4.3.3								



	Análisis de requisitos y planificación	Revisión del análisis	Diseño	Revisión del diseño	Codificación	Revisión de codificación	Pruebas	Mantenimiento
A.4.4.1								
A.4.4.2								
A.4.4.3								
A.5.1.1								
A.5.1.2	X							
A.5.1.3								
A.5.1.4		X		X		X	X	
A.5.1.5							X	
A.5.1.6							X	
A.5.1.7	X							
A.5.2.1			X					
A.5.2.2	X	X	X	X	X	X	X	X
A.5.2.3	X	X	X	X	X	X	X	X
A.5.3.1			X			X		
A.5.3.2								X
A.5.4.1								
A.5.4.2								
A.5.4.3								

- Tabla 22: Actividades y tareas -

3.2.1.3.- ACTIVIDADES Y TAREAS QUE NO SE REALIZAN

1.2.3 Plan de transición del sistema: Se prescinde de esta tarea debido no se sustituye ningún sistema anterior.

1.2.8 Plan de integración: La integridad del software frente a otros se realiza y se verifica en otra tarea.

1.3.4 Conservación de registros: No se realizan históricos sobre documentos de planificación sino que se van modificando y actualizando.

2.1.2 Formulación de métodos potenciales: No se necesita generar documentos ni informes acerca de la decisión de adquirir un sistema ya que se debe generar un software, no adquirirlo.

2.1.3 Estudio de las conductas de viabilidad: Ya conocemos los beneficios del ciclo de vida seleccionado ya que se obtiene de la revisión y del análisis y planificación de requisitos. Además está claro que el sistema se va a crear y no a adquirir.

2.1.4 Refinar y finalizar la idea o necesidad: Dado que se prescinde de las dos tareas anteriores no es necesario realizar esta.



2.3 Actividades importadas de software: Debido a que no se importa nada, no es necesario realizar ninguna de las tareas de este apartado.

4.2.3 Mantener el soporte para el Log de petición: Esta tarea no se realiza porque no se creará ningún archivo de Log.

4.3 Actividades de mantenimiento: Este tipo de actividades se realizan en otras tareas; por ello, se prescinde de ella en este apartado.

4.4 Retirada de actividades: Como se ha comentado anteriormente, al no sustituirse ningún sistema anterior estas tareas no son necesarias.

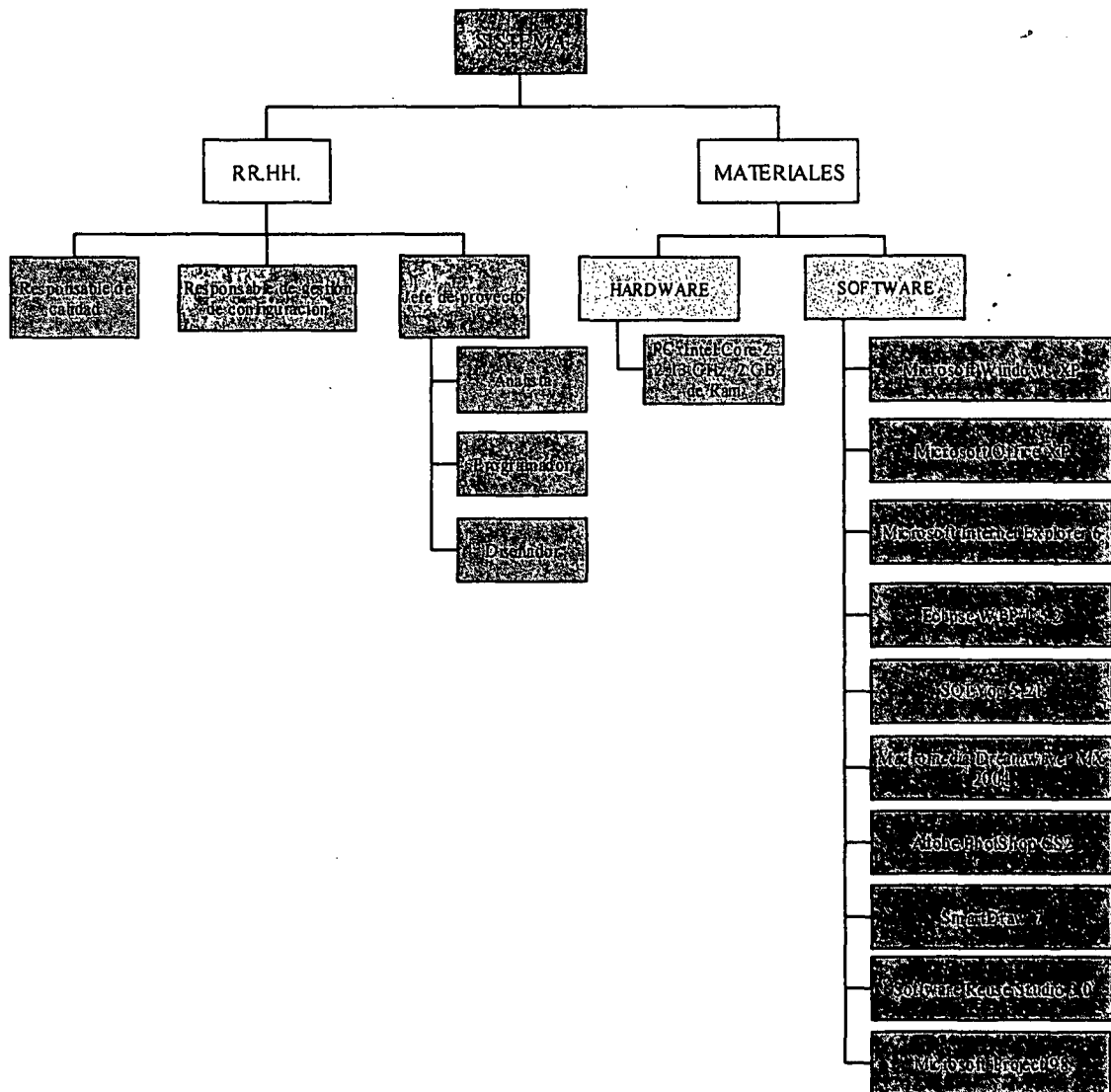
5.1.1 Revisión de conductas: Se realizan durante la planificación del proyecto y la revisión posterior de la misma.

5.1.3 Auditorias de conductas: Como solo se crea un producto software, no es necesario llevar a cabo esta tarea.

5.4 Actividades de entrenamiento: No se desarrollará ninguna tarea de esta actividad ya que las pruebas realizadas en otras etapas del ciclo de vida se han realizado ya con la base de datos implementada. Si es necesario añadir nuevos datos dependerá del propietario del sistema.

3.2.2.- RESOURCE BREAKDOWN STRUCTURE (RBS)

El RBS es un árbol jerárquico en el que se representan los recursos humanos y materiales de los que está formado un proyecto



- Figura 11: Alcance WBS -

Todos los componentes del equipo son nuevos en su labor y tienen poca experiencia en el desarrollo de proyectos de esta envergadura.

El programador cuenta con un nivel medio-alto en programación, pero al ser recién salido de la universidad no tiene conocimientos elevados en J2EE.

El jefe de proyecto en general puede desempeñar casi todas las tareas, pero principalmente supervisa todo lo que ocurre en el proyecto y es el nexo de unión entre los componentes del equipo y los clientes.

El responsable de calidad se encargará principalmente de realizar las siguientes tareas:

- Realizar el seguimiento y control de la calidad del sistema en desarrollo.



- Realizar el seguimiento y control de la calidad del proceso, las métricas y los procedimientos seguidos en el proyecto.
- Indicar que acciones correctivas deben tomarse en caso de encontrar inconsistencias o incumplimientos de la calidad.
- Asegurar que el sistema desarrollado cumple con determinadas propiedades

El responsable de gestión de configuración se encargará principalmente de realizar las siguientes tareas:

- Identificar los elementos del proyecto que deben estar bajo configuración.
- Restringir los cambios a dichos elementos.
- Auditar los cambios a esos elementos.
- Definir y gestionar la configuración de esos elementos.

El analista se encargará de las siguientes tareas:

- Conocer en detalle los requerimientos del sistema profundizando en el análisis de algunos CU.
- Entender claramente los requerimientos del Sistema y la metodología de desarrollo basada en CU.
- Verificar la trazabilidad entre los distintos modelos.
- Definir junto con el arquitecto, el diseño del sistema.
- Hacer el documento de requerimientos y la especificación de requerimientos para validar con el cliente.
- Realizar junto con el administrador estimaciones de tamaño mediante el cálculo de Puntos de Función.

El diseñador desempeñará las siguientes tareas:

- Transformar los requerimientos en el diseño de lo que debe ser el sistema.
- Desarrollar una arquitectura robusta para el sistema.



- Adaptar el diseño para que se corresponda con el entorno de implementación, diseñando de forma que la implementación tenga buena ejecución.
- Para lograr estos objetivos es necesario conocer el alcance del sistema, las herramientas de desarrollo y los recursos disponibles.

El programador se dedicará a realizar las siguientes tareas:

- Definir la organización del código en términos de implementación de subsistemas organizados en capas.
- Implementar clases y objetos términos de componentes (archivos fuentes, binarios, ejecutables y otros).

Y referente a las pruebas, el programador desempeñará las siguientes tareas:

- Verificar la interacción entre componentes.
- Verificar la correcta integración de todos los componentes del sistema.
- Verificar que todos los requerimientos han sido correctamente implementados.
- Identificar y asegurar que los defectos sean corregidos antes de la liberación del la versión

En cuanto a los recursos materiales, los más importantes se encuentran explicados en el apartado 2.2. *Herramientas y tecnologías utilizadas*.

3.2.3.- PRODUCTOS A OBTENER EN CADA TAREA (PBS)

ANÁLISIS DE REQUISITOS Y PLANIFICACIÓN

1.1 Iniciación de actividades del proyecto

- 1.1.1 Creación del ciclo de vida: Ciclo de vida
- 1.1.2 Realización de estimaciones: Resultados de la estimación
- 1.1.3 Distribución de los recursos del proyecto: Plan de distribución de actividades y recursos.
- 1.1.4 Definición de métricas: Métricas



1.2 Planificación de actividades del proyecto

- 1.2.2 Plan de configuración del desarrollo: Documentación sobre la configuración.
- 1.2.4 Plan de instalación: Manual de instalación.
- 1.2.5 Plan de documentación: Documentación sobre la información disponible.
- 1.2.6 Pruebas: Documento de requisitos.
- 1.2.7 Desarrollo del plan del proyecto: Informe sobre la asignación de responsabilidades en el proyecto.

1.3 Visualización y control de las actividades del proyecto

- 1.3.1 Gestión de riesgos: Informe de riesgos.
- 1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

2.1 Actividades de exploración conceptual

- 2.1.1 Identificación de ideas o necesidades: Informe de las ideas generadas.

3.1 Actividades requeridas

- 3.1.1 Definir y desarrollar requisitos software: Documentación de los requisitos del usuario y de los propuestos por los desarrolladores.

5.1 Evaluación de actividades

- 5.1.2 Crear una matriz de rastreo: Implementación de la matriz de rastreo e informe.
- 5.1.7 Notificar los resultados de evaluación: Informe de los resultados obtenidos en la evaluación

5.2 Actividades de gestión de configuración de software

- 5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.
- 5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

REVISIÓN DEL ANÁLISIS DE REQUISITOS Y PLANIFICACIÓN

1.2 Planificación de actividades del proyecto

- 1.2.5 Plan de documentación: Documentación sobre la información disponible.



1.2.7 Desarrollo del plan del proyecto: Informe sobre la asignación de responsabilidades en el proyecto.

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

1.3.3 Identificar las mejoras necesarias del ciclo de vida: Informe de evaluación de mejora del ciclo de vida.

1.3.5 Recogida y análisis de datos de la métrica: Análisis de evaluación de la métrica empleada.

3.1 Actividades requeridas

3.1.1 Definir y desarrollar requisitos software: Documentación de los requisitos del usuario y de los propuestos por los desarrolladores.

5.1 Evaluación de actividades

5.1.4 Desarrollo de procesos de prueba: Tabla de características de las pruebas realizadas.

5.2 Actividades de gestión de configuración de software

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

DISEÑO

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

2.2 Sistema de asignación de actividades

2.2.1 Analizar funciones: Análisis del diseño del software y hardware y arquitectura a seguir para su desarrollo.

2.2.2 Desarrollo de la arquitectura del sistema: Determinación de la arquitectura del sistema a seguir e informe de desarrollo.

2.2.3 Descomponer requisitos del sistema: Informe desglosado de los requisitos hardware y software humano.



3.1 Actividades requeridas

3.1.1 Definir y desarrollar requisitos software: Documentación de los requisitos del usuario y de los propuestos por los desarrolladores.

3.1.2 Definir requisitos de interfaz: Listado con los requisitos especificados para el desarrollo del software.

3.2 Diseño de actividades

3.2.1 Ejecutar el diseño de arquitectura: Informe con la identificación de los componentes software y sus estructuras.

3.2.2 Diseñar la base de datos: Diseño del modelo E/R, relacional, álgebra relacional y normalización de la base de datos.

3.2.3 Diseñar interfaces: Interfaces que se emplearán en el software que se desarrolla.

5.2 Actividades de gestión de configuración de software

5.2.1 Desarrollar identificaciones de configuración: Documentos que recogen la configuración o los identificadores de configuración incluyendo las líneas base.

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

5.3 Documentación de las actividades de desarrollo

5.3.1 Implementación de la documentación: Documentación obtenida tras el análisis. Casos de uso, diagramas de actividad, diagrama de clases, diagramas de secuencia, diagrama E/R y modelo relacional de la base de datos.

REVISIÓN DEL DISEÑO

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

1.3.3 Identificar las mejoras necesarias del ciclo de vida: Informe de evaluación de mejora del ciclo de vida.

2.2 Sistema de asignación de actividades

2.2.2 Desarrollo de la arquitectura del sistema: Determinación de la arquitectura del sistema a seguir e informe de desarrollo.



3.1 Actividades requeridas

3.1.1 Definir y desarrollar requisitos software: Documentación de los requisitos del usuario y de los propuestos por los desarrolladores.

3.1.3 Priorizar e integrar requisitos software: Informe de los requisitos integrados en el software y documentación de la decisión tomada.

3.2 Diseño de actividades

3.2.2 Diseñar la base de datos: Diseño del modelo E/R, relacional, álgebra relacional y normalización de la base de datos.

5.1 Evaluación de actividades

5.1.4 Desarrollo de procesos de prueba: Tabla de características de las pruebas realizadas.

5.2 Actividades de gestión de configuración de software

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

CODIFICACIÓN

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

3.2 Diseño de actividades

3.2.4 Ejecutar los diseños detallados: Codificación del software.

3.3 Implementación de actividades

3.3.1 Crear código ejecutable: Código fuentes agrupado por unidades (modulado)

3.3.3 Integración de la ejecución: Código fuente de la base de datos y documentación planificada en el software.



5.2 Actividades de gestión de configuración de software

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

REVISIÓN DE CODIFICACIÓN

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

3.3 Implementación de actividades

3.3.1 Crear código ejecutable: Código fuentes agrupado por unidades (modulado).

3.3.2 Crear la documentación operativa: Documentación del diseño detallado y requisitos de la interfaz software.

3.3.3 Integración de la ejecución: Código fuente de la base de datos y documentación planificada en el software.

5.1 Evaluación de actividades

5.1.4 Desarrollo de procesos de prueba: Tabla de características de las pruebas realizadas.

5.2 Actividades de gestión de configuración de software

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

5.3 Documentación de las actividades de desarrollo

5.3.1 Implementar la documentación: Actualización de los documentos generados en la fase de diseño.



PRUEBAS

1.2 Planificación de actividades del proyecto

1.2.1 Plan de evaluaciones: Documentación sobre las pruebas de evaluación del software.

1.2.6 Pruebas: Documento de requisitos.

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

4.1 Instalación de actividades

4.1.1 Distribuir el software: Envío de los paquetes que contienen la información necesaria para la instalación y el software.

4.1.3 Aceptar el software en el entorno operacional: Resultados de las pruebas de instalación del software.

5.1 Evaluación de actividades

5.1.4 Desarrollo de procesos de prueba: Tabla de características de las pruebas realizadas.

5.1.5 Crear pruebas de datos: Resultados de las pruebas ejecutadas en la base de datos.

5.1.6 Ejecutar pruebas: Informe de anomalías producidas tras las pruebas y de resultados de la realización de las comparaciones pertinentes.

5.2 Actividades de gestión de configuración de software

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

MANTENIMIENTO

1.3 Visualización y control de las actividades del proyecto

1.3.1 Gestión de riesgos: Informe de riesgos.

1.3.2 Gestión del proyecto: Comparación sobre la realidad y lo estimado.

4.1 Instalación de actividades

4.1.2 Instalar el software: Instalación del software.



4.2 Operación y soporte de actividades

4.2.2 Proporcionar asistencia técnica y consultoría: Creación del soporte de asistencia y plan de asistencia.

5.2 Actividades de gestión de configuración de software

5.2.2 Realizar control de configuraciones: Establecimiento de la configuración del software y creación de documento del mismo.

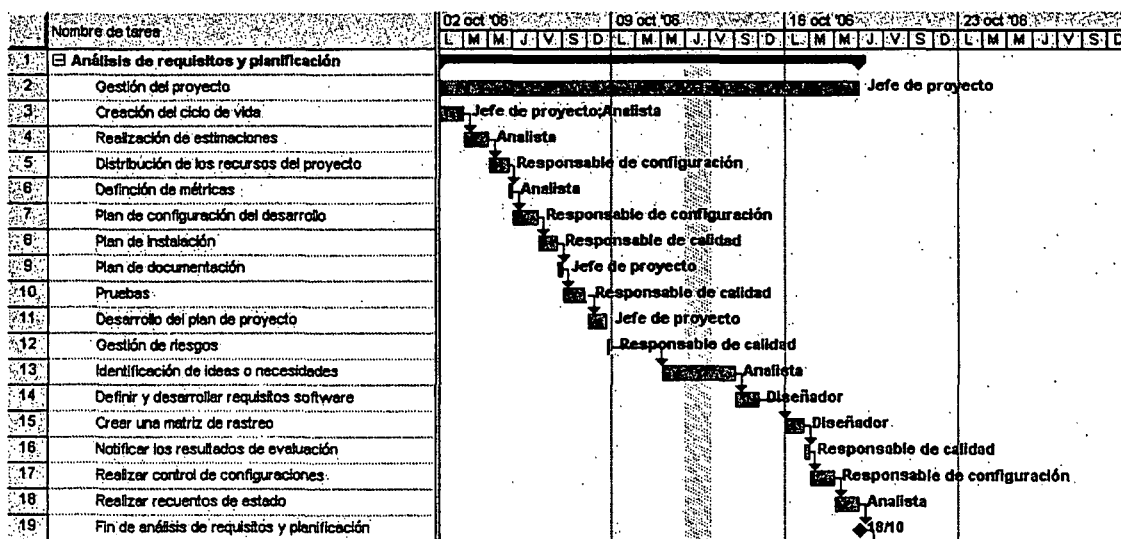
5.2.3 Realizar recuentos de estados: Actualización de la configuración de estado.

5.3 Documentación de las actividades de desarrollo

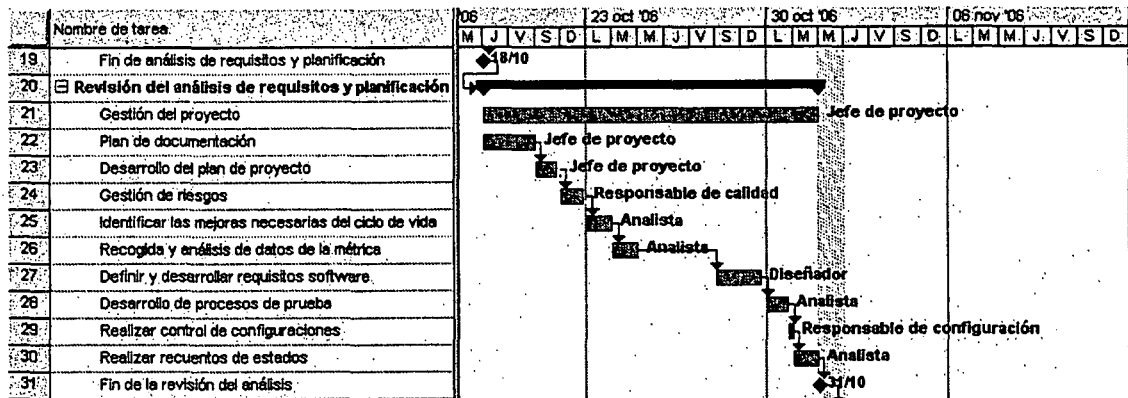
5.3.2 Producir y distribuir la documentación: Documento que recoge toda la documentación.

3.2.4.- PLANIFICACIÓN

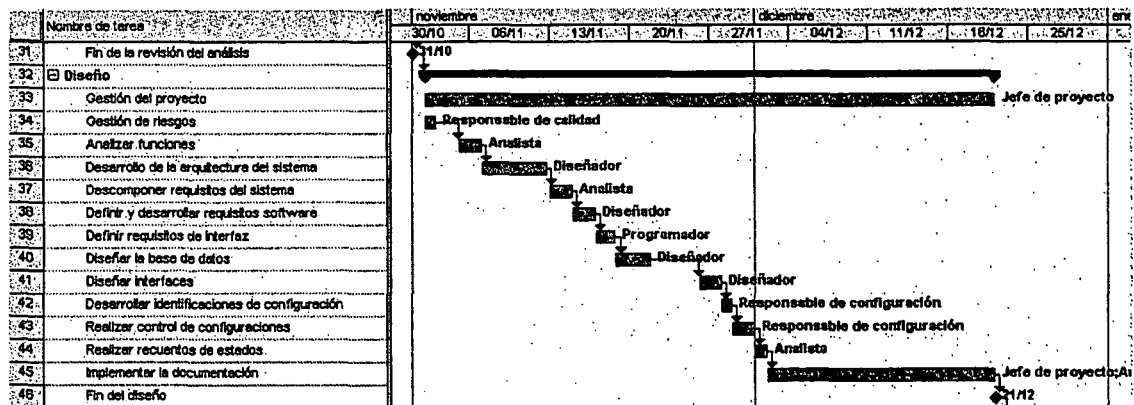
A continuación, se muestra la planificación del proyecto a través de un diagrama Gantt, en el cual se refleja cada una de las tareas realizadas en el proyecto, su duración y los recursos asignados.



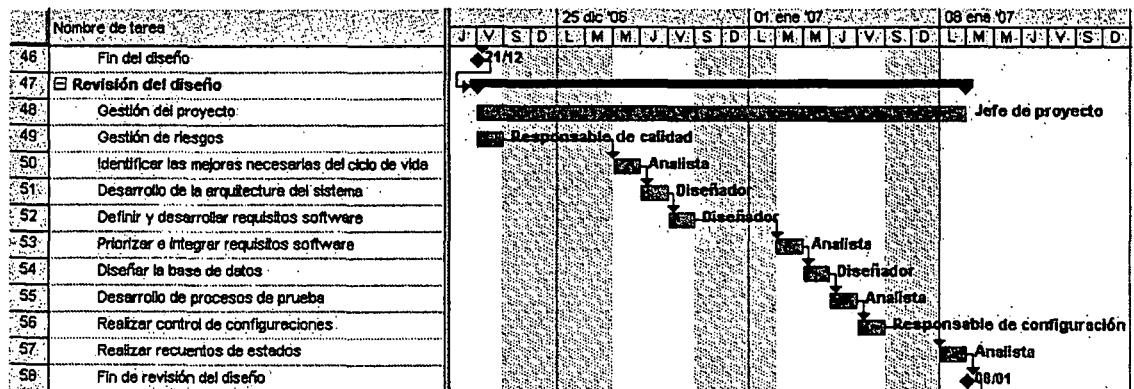
- Figura 12: Diagrama de Gantt 1 -



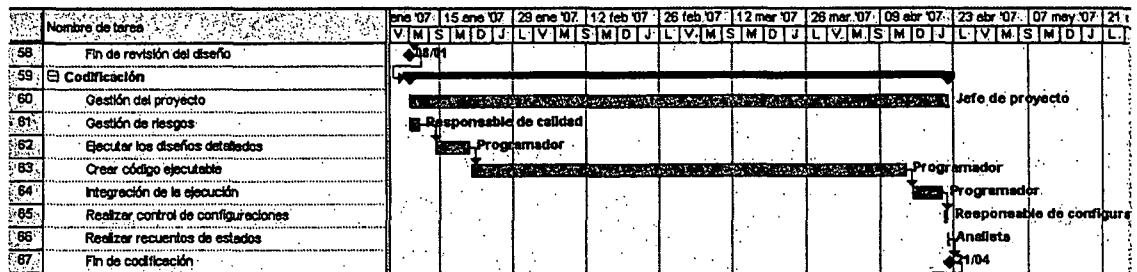
- Figura 13: Diagrama de Gantt 2 -



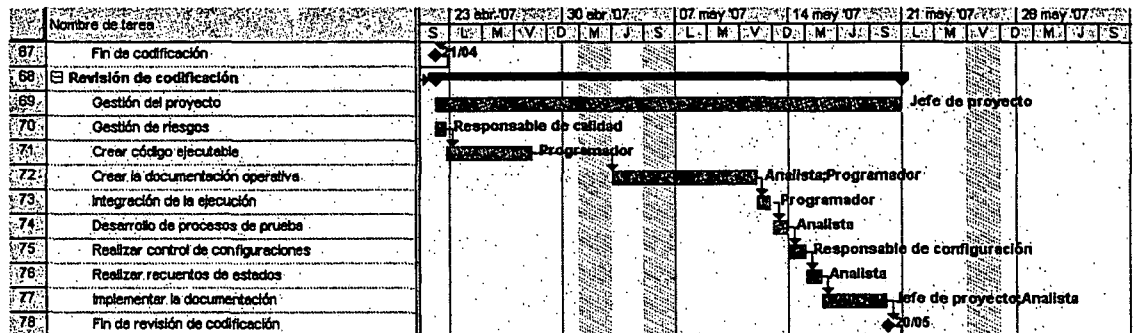
- Figura 14: Diagrama de Gantt 3 -



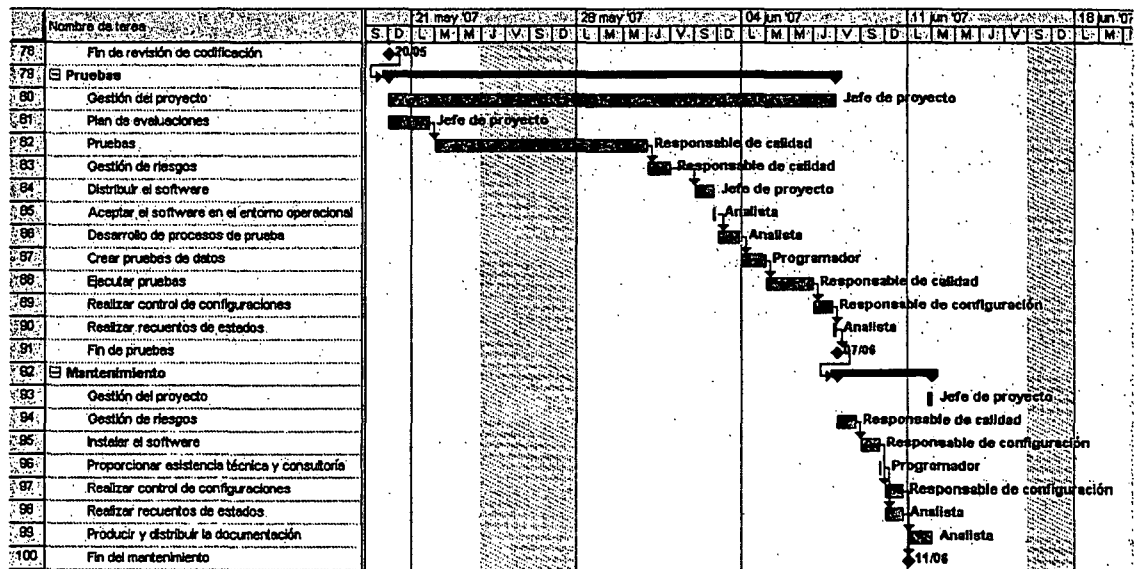
- Figura 15: Diagrama de Gantt 4 -



- Figura 16: Diagrama de Gantt 5 -



- Figura 17: Diagrama de Gantt 6 -



- Figura 18: Diagrama de Gantt 7 -



3.2.5.- ASIGNACIÓN DE RECURSOS

En la siguiente tabla, se refleja la asignación de los diferentes recursos que conforman el proyecto a las tareas, así como la duración de las mismas.

De esta forma se describe con mayor claridad el diagrama de Gantt.

ACTIVIDAD/TAREA	DURACIÓN	INICIO	FIN	PERSONAL ASIGNADO
<i>Análisis de requisitos y planificación</i>	16 días	02/10/2006	18/10/2006	
<i>Gestión del proyecto</i>	16 días	02/10/2006	18/10/2006	Jefe de proyecto
<i>Creación del ciclo de vida</i>	1 día	02/10/2006	02/10/2006	Jefe de proyecto; Analista
<i>Realización de estimaciones</i>	1 día	03/10/2006	03/10/2006	Analista
<i>Distribución de los recursos del proyecto</i>	0,5 días	04/10/2006	04/10/2006	Responsable de configuración
<i>Definición de métricas</i>	0,5 días	04/10/2006	04/10/2006	Analista
<i>Plan de configuración del desarrollo</i>	1 día	05/10/2006	05/10/2006	Responsable de configuración
<i>Plan de instalación</i>	0,5 días	06/10/2006	06/10/2006	Responsable de calidad
<i>Plan de documentación</i>	0,5 días	06/10/2006	06/10/2006	Jefe de proyecto
<i>Pruebas</i>	1 día	07/10/2006	07/10/2006	Responsable de calidad
<i>Desarrollo del plan de proyecto</i>	0,5 días	08/10/2006	08/10/2006	Jefe de proyecto
<i>Gestión de riesgos</i>	0,5 días	08/10/2006	08/10/2006	Responsable de calidad
<i>Identificación de ideas o necesidades</i>	2 días	11/10/2006	13/10/2006	Analista
<i>Definir y desarrollar requisitos software</i>	2 días	14/10/2006	15/10/2006	Diseñador
<i>Crear una matriz de rastreo</i>	0,5 días	16/10/2006	16/10/2006	Diseñador
<i>Notificar los resultados de evaluación</i>	0,5 días	16/10/2006	16/10/2006	Responsable de calidad
<i>Realizar control de configuraciones</i>	1 día	17/10/2006	17/10/2006	Responsable de configuración



ACTIVIDAD / TAREA	DURACIÓN	FECHA DE INICIO	FECHA DE FIN	PERSONAL ASIGNADO
Realizar recuentos de estado	1 día	18/10/2006	18/10/2006	Analista
Fin de análisis de requisitos y planificación	0 días	18/10/2006	18/10/2006	
Revisión del análisis de requisitos y planificación	12 días	19/10/2006	31/10/2006	
Gestión del proyecto	12 días	19/10/2006	31/10/2006	Jefe de proyecto
Plan de documentación	2 días	19/10/2006	20/10/2006	Jefe de proyecto
Desarrollo del plan de proyecto	1 día	21/10/2006	21/10/2006	Jefe de proyecto
Gestión de riesgos	1 día	22/10/2006	22/10/2006	Responsable de calidad
Identificar las mejoras necesarias del ciclo de vida	1 día	25/10/2006	25/10/2006	Analista
Recogida y análisis de datos de la métrica	1 día	26/10/2006	26/10/2006	Analista
Definir y desarrollar requisitos software	2 días	28/10/2006	29/10/2006	Diseñador
Desarrollo de procesos de prueba	0,5 días	30/10/2006	30/10/2006	Analista
Realizar control de configuraciones	0,5 días	30/10/2006	30/10/2006	Responsable de configuración
Realizar recuentos de estados	1 día	31/10/2006	31/10/2006	Analista
Fin de la revisión del análisis	0 días	31/10/2006	31/10/2006	
Diseño	44 días	02/11/2006	22/12/2006	
Gestión del proyecto	44 días	02/11/2006	22/12/2006	Jefe de proyecto
Gestión de riesgos	1 día	02/11/2006	02/11/2006	Responsable de calidad
Analizar funciones	2 días	05/11/2006	06/11/2006	Analista
Desarrollo de la arquitectura del sistema	6 días	07/11/2006	12/11/2006	Diseñador
Descomponer requisitos del sistema	2 días	13/11/2006	14/11/2006	Analista
Definir y desarrollar requisitos software	2 días	15/11/2006	16/11/2006	Diseñador
Definir requisitos de interfaz	2 días	17/11/2006	18/11/2006	Programador
Diseñar la base de datos	4 días	18/11/2006	21/11/2006	Diseñador



ACTIVIDAD / TAREA	DURACIÓN	INICIO	FIN	PERSONAL ASIGNADO
<i>Diseñar interfaces</i>	2 días	26/11/2006	27/11/2006	Diseñador
<i>Desarrollar identificaciones de configuración</i>	1 día	28/11/2006	28/11/2006	Responsable de configuración
<i>Realizar control de configuraciones</i>	2 días	29/11/2006	30/11/2006	Responsable de configuración
<i>Realizar recuentos de estados</i>	1 día	01/12/2006	01/12/2006	Analista
<i>Implementar la documentación</i>	17 días	02/12/2006	21/12/2006	Jefe de proyecto; Analista
<i>Fin del diseño</i>	0 días	21/12/2006	21/12/2006	
<i>Revisión del diseño</i>	10 días	22/12/2006	09/01/2007	
<i>Gestión del proyecto</i>	10 día	22/12/2006	09/01/2007	Jefe de proyecto
<i>Gestión de riesgos</i>	1 día	22/12/2006	22/12/2006	Responsable de calidad
<i>Identificar las mejoras necesarias del ciclo de vida</i>	1 día	28/12/2006	28/12/2006	Analista
<i>Desarrollo de la arquitectura del sistema</i>	1 día	29/12/2006	29/12/2006	Diseñador
<i>Definir y desarrollar requisitos software</i>	1 día	02/01/2007	02/01/2007	Diseñador
<i>Priorizar e integrar requisitos software</i>	1 día	03/01/2007	03/01/2007	Analista
<i>Diseñar la base de datos</i>	1 día	04/01/2007	04/01/2007	Diseñador
<i>Desarrollo de procesos de prueba</i>	1 día	05/01/2007	05/01/2007	Analista
<i>Realizar control de configuraciones</i>	1 día	08/01/2007	08/01/2007	Responsable de configuración
<i>Realizar recuentos de estados</i>	1 día	09/01/2007	09/01/2007	Analista
<i>Fin de revisión del diseño</i>	0 días	09/01/2007	09/01/2007	
<i>Codificación</i>	95 días	10/01/2007	21/04/2007	
<i>Gestión del proyecto</i>	95 días	10/01/2007	21/04/2007	Jefe de proyecto
<i>Gestión de riesgos</i>	2 días	10/01/2007	11/01/2007	Responsable de calidad
<i>Ejecutar los diseños detallados</i>	7 días	14/01/2007	20/01/2007	Programador
<i>Crear código ejecutable</i>	77 días	21/01/2007	14/04/2007	Programador



ACTIVIDAD/ TAREA	DURACION	INICIO	FIN	PERSONAL ASIGNADO
<i>Integración de la ejecución</i>	6 días	15/04/2007	20/04/2007	Programador
<i>Realizar control de configuraciones</i>	0,5 días	21/04/2007	21/04/2007	Responsable de configuración
<i>Realizar recuentos de estados</i>	0,5 días	21/04/2007	21/04/2007	Analista
<i>Fin de codificación</i>	0 días	21/04/2007	21/04/2007	
<i>Revisión de codificación</i>	25 días	22/04/2007	20/05/2007	
<i>Gestión del proyecto</i>	25 día	22/04/2007	20/05/2007	Jefe de proyecto
<i>Gestión de riesgos</i>	1 día	22/04/2007	22/04/2007	Responsable de calidad
<i>Crear código ejecutable</i>	7 días	24/04/2007	30/04/2007	Programador
<i>Crear la documentación operativa</i>	7 días	03/05/2007	11/05/2007	Analista;Programador
<i>Integración de la ejecución</i>	1 día	12/05/2007	12/05/2007	Programador
<i>Desarrollo de procesos de prueba</i>	1 día	13/05/2007	13/05/2007	Analista
<i>Realizar control de configuraciones</i>	1 día	14/05/2007	14/05/2007	Responsable de configuración
<i>Realizar recuentos de estados</i>	1 día	15/05/2007	15/05/2007	Analista
<i>Implementar la documentación</i>	5 días	16/05/2007	20/05/2007	Jefe de proyecto;Analista
<i>Fin de revisión de codificación</i>	0 días	20/05/2007	20/05/2007	
<i>Pruebas</i>	40 días	20/05/2007	07/06/2007	
<i>Gestión del proyecto</i>	40 día	20/05/2007	07/06/2007	Jefe de proyecto
<i>Plan de evaluaciones</i>	2 días	20/05/2007	21/05/2007	Jefe de proyecto
<i>Pruebas</i>	5 días	22/05/2007	30/05/2007	Responsable de calidad
<i>Gestión de riesgos</i>	1 día	31/05/2007	31/05/2007	Responsable de calidad
<i>Distribuir el software</i>	0,5 días	02/06/2007	02/06/2007	Jefe de proyecto
<i>Aceptar el software en el entorno operacional</i>	0,5 días	02/06/2007	02/06/2007	Analista
<i>Desarrollo de procesos de prueba</i>	1 día	03/06/2007	03/06/2007	Analista



ACTIVIDAD / TAREA	DURACIÓN	INICIO	FIN	PERSONAL ASIGNADO
Crear pruebas de datos	1 día	04/06/2007	04/06/2007	Programador
Ejecutar pruebas	2 días	05/06/2007	06/06/2007	Responsable de calidad
Realizar control de configuraciones	0,5 días	07/06/2007	07/06/2007	Responsable de configuración
Realizar recuentos de estados	0,5 días	07/06/2007	07/06/2007	Analista
Fin de pruebas	0 días	07/06/2007	07/06/2007	
Mantenimiento	4 días	08/06/2007	11/06/2007	
Gestión del proyecto	4 días	08/06/2007	11/06/2007	Jefe de proyecto
Gestión de riesgos	0,5 días	08/06/2007	08/06/2007	Responsable de calidad
Instalar el software	0,5 días	09/06/2007	09/06/2007	Responsable de configuración
Proporcionar asistencia técnica y consultoría	0,5 días	09/06/2007	09/06/2007	Programador
Realizar control de configuraciones	0,5 días	10/06/2007	10/06/2007	Responsable de configuración
Realizar recuentos de estados	0,5 días	10/06/2007	10/06/2007	Analista
Producir y distribuir la documentación	1 día	11/06/2007	11/06/2007	Analista
Fin del mantenimiento	0 días	11/06/2007	11/06/2007	

- Tabla 23: Asignación de recursos -



3.2.6.- ESTIMACIÓN DE COSTES

En la siguiente tabla se refleja el salario neto de los empleados:

EMPLEADO	% TRATAMIENTO	TASA ESTÁNDAR	TASA HORAS EXTRA
<i>Analista</i>	100%	35 €/hora	38 €/hora
<i>Programador</i>	100%	32 €/hora	35 €/hora
<i>Diseñador</i>	100%	32 €/hora	35 €/hora
<i>Responsable de calidad</i>	100%	38 €/hora	40 €/hora
<i>Responsable de configuración</i>	100%	38 €/hora	40 €/hora
<i>Jefe de proyecto</i>	100%	40 €/hora	42 €/hora

- Tabla 24: Asignación de presupuestos -

Finalmente se representa cada fase del proyecto, con el honorario que ha recibido cada empleado y el total de horas trabajadas durante dichas fases.

Se debe tener en cuenta que el horario de trabajo es el siguiente:

- De lunes a viernes: Desde 17:00 h, hasta 21:30 h.
- De sábado a domingo: De 10:00 h a 14:00h, y de 17:00 a 21:00.

FASE/EMPLEADOS	COSTE (€)	HORAS
Análisis de requisitos y planificación	3.070,75	83,5
<i>Jefe de proyecto</i>	790	19,75
<i>Responsable de configuración</i>	427,5	11,25
<i>Responsable de calidad</i>	627	16,5
<i>Analista</i>	866,25	24,75
<i>Diseñador</i>	360	11,25
<i>Programador</i>	0	0
Revisión del análisis de requisitos y planificación	2.492,75	68
<i>Jefe de proyecto</i>	1.040	26
<i>Responsable de configuración</i>	85,5	2,25
<i>Responsable de calidad</i>	304	8
<i>Analista</i>	551,25	15,75
<i>Diseñador</i>	512	16
<i>Programador</i>	0	0
Diseño	12.382,5	344,5
<i>Jefe de proyecto</i>	4.400	110
<i>Responsable de configuración</i>	5133	13,5
<i>Responsable de calidad</i>	171	4,5



BASE EMPLEADOS	COSTE(C)	HORAS
Analista	4.322,5	123,5
Diseñador	2.576	80,5
Programador	400	12,5
Revisión del diseño	1.584	45
Jefe de proyecto	180	4,5
Responsable de configuración	171	4,5
Responsable de calidad	171	4,5
Analista	630	18
Diseñador	432	13,5
Programador	0	0
Codificación	24.702	766
Jefe de proyecto	500	12,5
Responsable de configuración	152	4
Responsable de calidad	342	9
Analista	140	4
Diseñador	0	0
Programador	23.568	736,5
Revisión de codificación	6.367,5	182
Jefe de proyecto	1.360	34
Responsable de configuración	171	4,5
Responsable de calidad	304	8
Analista	2.292,5	65,5
Diseñador	0	0
Programador	2.240	70
Pruebas	2.936,25	78
Jefe de proyecto	840	21
Responsable de configuración	85,5	2,25
Responsable de calidad	1.368	36
Analista	498,75	14,25
Diseñador	0	0
Programador	114	4,5
Mantenimiento	905	25
Jefe de proyecto	90	2,25
Responsable de configuración	304	8
Responsable de calidad	85,5	2,25
Analista	297,5	8,5
Diseñador	0	0
Programador	128	4
TOTAL	54.440,75	1.592

- Tabla 25: Estimación del coste -



4.- ANÁLISIS Y DISEÑO DEL PROBLEMA

4.1.- DIAGRAMA DE CASOS DE USO

Mediante este diagrama se proporciona una visión de los requisitos funcionales del proyecto que especifican el funcionamiento del sistema, reflejándose la interacción entre los actores y el propio sistema.

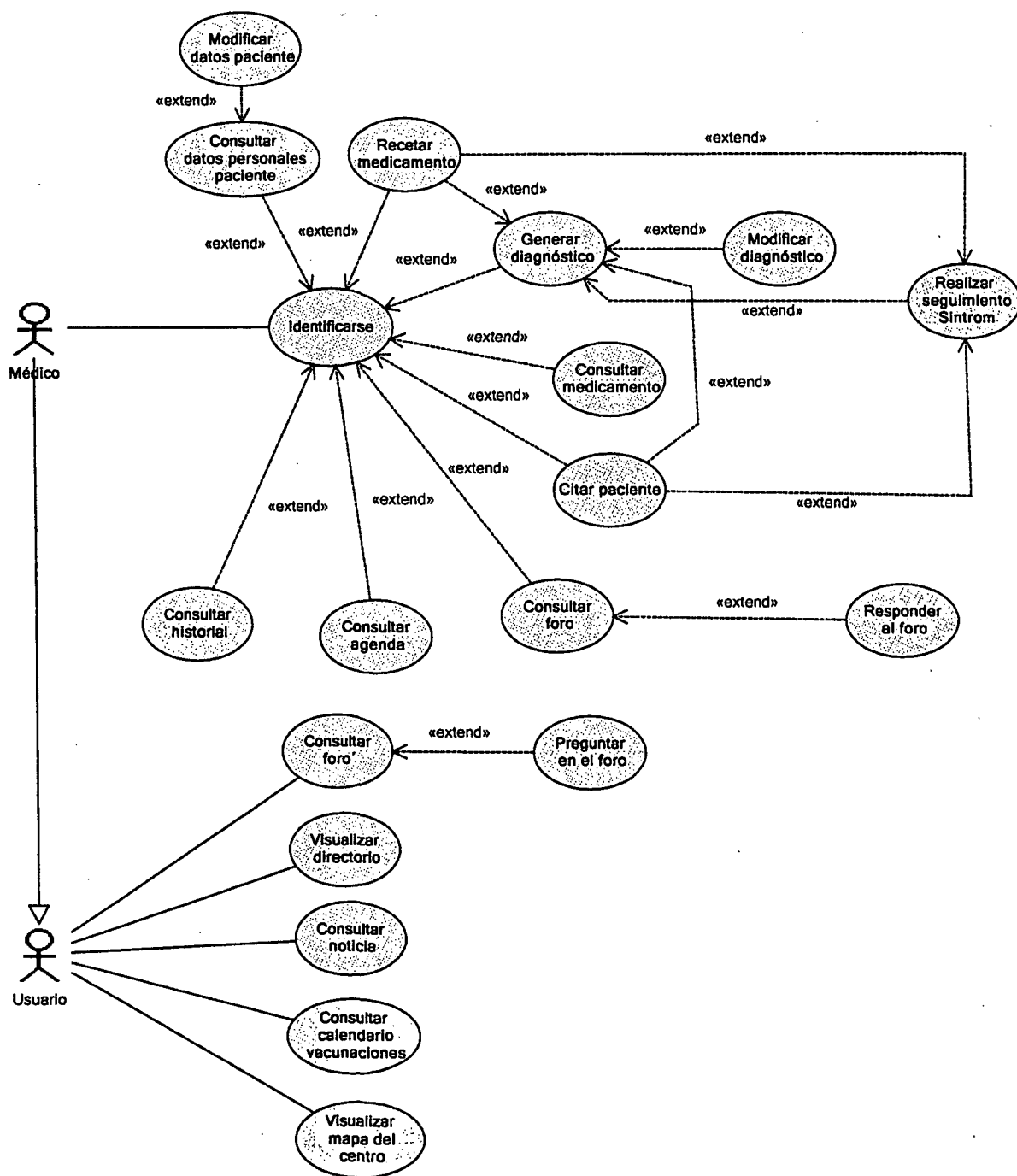
Los actores son las entidades externas que interactúan con el sistema, así en este caso, los actores son, médicos, administradores y usuarios.

En este diagrama, los casos de uso que son iguales pero son usados por distintos actores, se han representado con el mismo nombre y un apóstrofe, por ejemplo el caso de uso “*Identificarse*” del médico e “*Identificarse’*” del administrador, son realmente el mismo caso de uso.

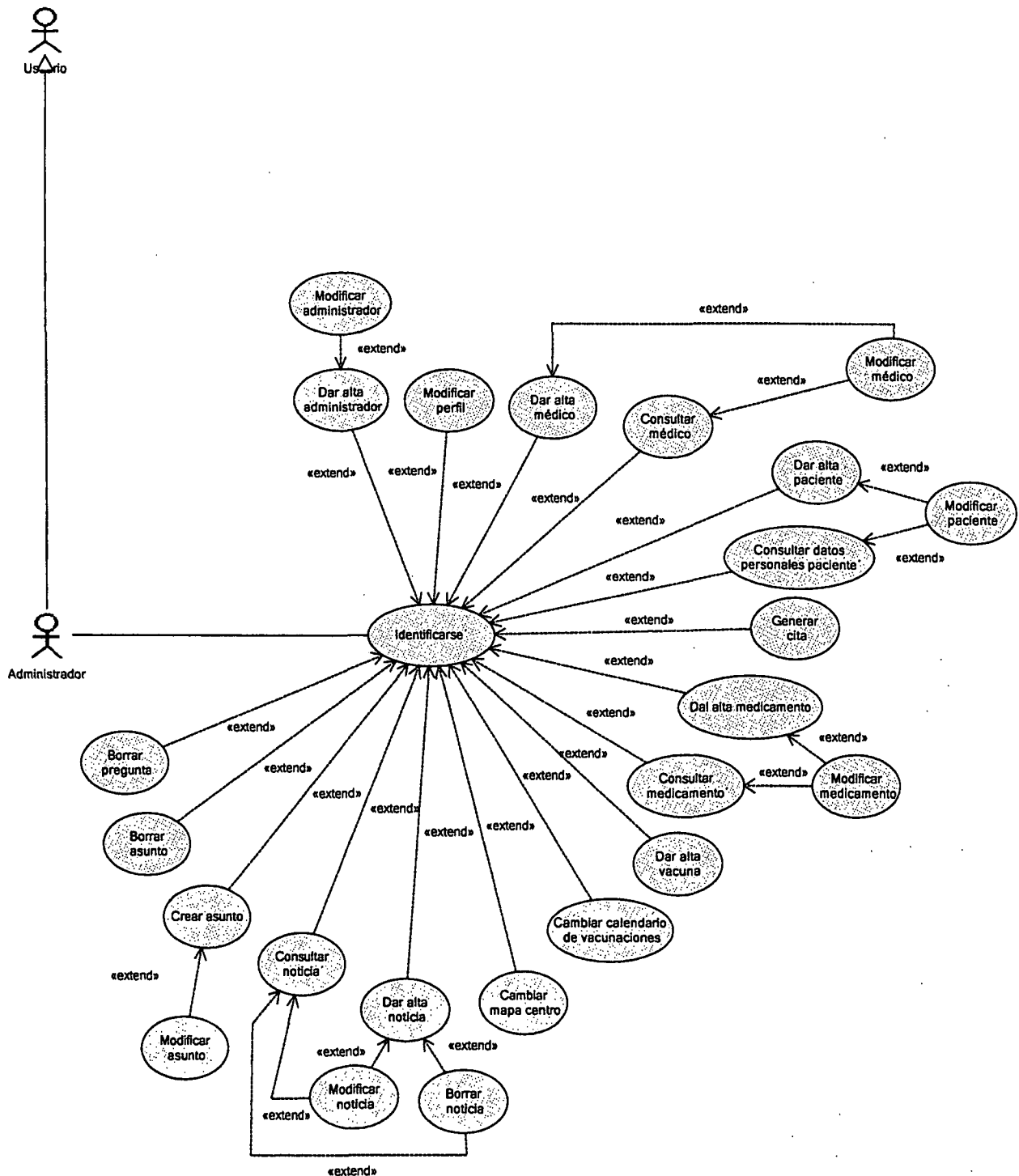
A parte de la relación de asociación, se ha hecho uso de la relación extend. A través de las relaciones extend se pretende mostrar que los casos de usos ligados hacia el que lleva el extend se pueden realizar o no una vez se haya realizado dicho caso.

Para que esto quede mas claro expondremos un ejemplo:

- **Extend:** Una vez que el médico se halla identificado en el sistema, puede realizar cualquier acción, “*Consultar medicamento*”, “*Consultar agenda*”...



- Figura 19: Diagrama de casos de uso 1 -



- Figura 20: Diagrama de casos de uso 2 -



4.1.1.- DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO

La descripción textual, es un documento narrativo en el que se describe la secuencia de eventos que se produce entre un actor y el sistema para llevar a cabo una tarea específica.

Nombre: Identificarse

Actores: Médico, Administrador.

Precondición:

Poscondición: Médico ó administrador identificado.

Objetivo: Identificar a un usuario como médico ó administrador para que pueda realizar las acciones propias de su condición.

Escenario básico: 1) Introducir usuario y contraseña.

2) Verificar su existencia en la BD.

Nombre: Consultar datos personales paciente

Actores: Médico, Administrador.

Precondición: Médico ó administrador identificado.

Poscondición: Datos paciente consultados.

Objetivo: Consultar los datos personales de un paciente.

Escenario básico: 1) Buscar al paciente seleccionando el/los criterios de búsqueda.

2) Visualizar listado de pacientes que corresponde con los criterios.

Opcional: Realizar de nuevo la búsqueda.

3) Seleccionar un paciente.

4) Consultar datos del paciente.

Nombre: Modificar datos paciente

Actores: Médico.

Precondición: Datos paciente consultados.

Poscondición: Datos paciente actualizados.

Objetivo: Modificar los datos de un paciente.

Escenario básico: 1) Modificar los datos pertinentes.

2) Actualizar BD.

Opcional: Añadir y/ó modificar alergias.



Nombre: Generar diagnóstico

Actores: Médico.

Precondición: Médico identificado y paciente dado de alta.

Poscondición: Diagnóstico generado.

Objetivo: Introducir el diagnóstico de un paciente que esta en la consulta.

Escenario básico: 1) Seleccionar número de historia del paciente.

2) Introducir datos necesarios para completar el diagnostico.

3) Actualizar BD.

4) Mostrar opciones realizar seguimiento de Sintrom (en caso de que se haya seleccionado), generar receta y citar paciente.

Opcional: Seleccionar alguna opción.

Nombre: Modificar diagnóstico

Actores: Médico.

Precondición: Diagnóstico generado.

Poscondición: Diagnóstico actualizado.

Objetivo: Modificar el diagnóstico que acabamos de generar.

Escenario básico: 1) Modificar los datos pertinentes.

2) Actualizar BD.

3) Mostrar opciones realizar seguimiento de Sintrom (en caso de que se haya seleccionado), generar receta y citar paciente.

Opcional: Seleccionar alguna opción.

Nombre: Realizar seguimiento del Sintrom.

Actores: Médico.

Precondición: Diagnostico generado.

Poscondición: Seguimiento realizado.

Objetivo: Realizar el seguimiento de Sintrom en un paciente.

Escenario básico: 1) Introducir datos necesarios para completar el seguimiento.

2) Actualizar BD.

3) Visualizar resultado.

Opcional: En caso de valor DTS $<0,5\text{mg}$ ó $>30\text{mg}$, introducir DTS adecuada y validar.

Opcional: Citar al paciente ó recetar medicamento.



Nombre: Recetar medicamento

Actores: Médico.

Precondición: Médico identificado y/o diagnóstico generado, y/o seguimiento Sintrom realizado.

Poscondición: Medicamento recetado.

Objetivo: Recetar un medicamento al paciente para tratar su dolencia.

Escenario básico: Repetir:

Opcional: Introducir número de historia del paciente, en caso de no venir desde un diagnóstico o seguimiento de Sintrom.

- 1) Buscar medicamento introduciendo el/los criterios de búsqueda.
- 2) Visualizar listado con los medicamentos que corresponden.

Opcional: Realizar de nuevo la búsqueda.

- 3) Seleccionar un medicamento.
- 4) Visualizar receta.

Opcional: Cambiar posología.

Opcional: Imprimir receta.

Nombre: Citar paciente

Actores: Médico.

Precondición: Médico identificado y/o diagnóstico generado, y/o seguimiento Sintrom realizado.

Poscondición: Paciente citado.

Objetivo: Citar a un paciente para acudir a la consulta.

Escenario básico: Repetir:

Opcional: Seleccionar número de historia del paciente, en caso de no venir desde un diagnóstico o seguimiento de Sintrom.

- 1) Introducir fecha y hora.
- 2) Visualizar cita.

Opcional: Imprimir cita.

Nombre: Consultar medicamento

Actores: Médico ó administrador.

Precondición: Médico ó administrador identificado.

Poscondición:

Objetivo: Consultar los datos de un medicamento.

- Escenario básico:**
- 1) Buscar el medicamento seleccionando el/los criterios de búsqueda.
 - 2) Visualizar listado de medicamentos que corresponden con los criterios.

Opcional: Realizar de nuevo la búsqueda.

- 3) Seleccionar un medicamento.
- 4) Consultar datos del medicamento.



Nombre: Consultar historial

Actores: Médico.

Precondición: Médico identificado.

Poscondición:

Objetivo: Consultar el historial médico de un paciente.

Escenario básico: 1) Buscar al paciente seleccionando el/los criterios de búsqueda.

2) Visualizar listado de pacientes que corresponden con los criterios.

Opcional: Realizar de nuevo la búsqueda.

3) Seleccionar un paciente.

4) Seleccionar un diagnóstico.

5) Visualizar diagnóstico.

Opcional: Visualizar el seguimiento de Sintrom en caso de existir.

Nombre: Consultar agenda

Actores: Médico.

Precondición: Médico identificado.

Poscondición:

Objetivo: Consultar las citas.

Escenario básico: 1) Introducir una fecha.

2) Visualizar listado de citas.

Opcional: Volver a introducir otra fecha.

Nombre: Consultar foro

Actores: Médico ó usuario.

Precondición:

Poscondición: Foro consultado.

Objetivo: Visualizar las respuestas y preguntas formuladas en el foro.

Escenario básico: 1) Seleccionar asunto.

2) Visualizar preguntas.

Opcional: Seleccionar pregunta y visualizar respuestas.

Nombre: Responder al foro

Actores: Médico.

Precondición: Foro consultado.

Poscondición: Respuesta formulada.

Objetivo: Responder a una pregunta formulada por un usuario.

Escenario básico: 1) Seleccionar pregunta, en caso de no haberla seleccionado con anterioridad.

2) Pulsar en responder y escribir la respuesta.

3) Actualizar foro.



Nombre: Preguntar en el foro

Actores: Usuario.

Precondición: Foro consultado.

Poscondición: Pregunta formulada.

Objetivo: Formular una pregunta para que sea respondida por un médico.

Escenario básico: 1) Pulsar en preguntar.
2) Escribir datos y pregunta.
3) Actualizar foro.

Nombre: Visualizar directorio

Actores: Usuario.

Precondición:

Poscondición: Datos del médico consultados.

Objetivo: Visualizar los datos de los médicos del centro.

Escenario básico: 1) Seleccionar un médico.
2) Visualizar datos personales.

Nombre: Consultar noticia

Actores: Usuario.

Precondición:

Poscondición:

Objetivo: Consultar las noticias de actualidad sobre en centro de salud.

Escenario básico: 1) Seleccionar título.
2) Visualizar noticia.

Nombre: Consultar calendario de vacunaciones

Actores: Usuario.

Precondición:

Poscondición:

Objetivo: Consultar el calendario general de vacunaciones.

Escenario básico: 1) Visualizar calendario de vacunaciones.

Nombre: Visualizar mapa del centro

Actores: Usuario.

Precondición:

Poscondición:

Objetivo: Visualizar el mapa del centro de salud.

Escenario básico: 1) Visualizar el mapa del centro de salud.



Nombre: Dar alta administrador

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Administrador dado de alta.

Objetivo: Dar de alta a otros administradores.

Escenario básico: 1) Introducir datos personales administrador.
2) Actualizar BD.

Nombre: Modificar administrador

Actores: Administrador.

Precondición: Administrador identificado y administrador dado de alta.

Poscondición: Administrador modificado.

Objetivo: Modificar los datos personales del administrador al que se acaba de dar de alta.

Escenario básico: 1) Modificar datos.
2) Actualizar BD.

Nombre: Modificar perfil

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Perfil modificado.

Objetivo: Modificar datos del administrador.

Escenario básico: 1) Modificar datos.
2) Actualizar BD.

Nombre: Dar alta médico

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Médico dado de alta.

Objetivo: Dar de alta a un médico.

Escenario básico: 1) Introducir datos personales del médico.
2) Actualizar BD.

Nombre: Modificar datos médico

Actores: Administrador.

Precondición: Médico consultado ó médico dado de alta.

Poscondición: Datos del médico actualizados.

Objetivo: Modificar los datos de un médico.

Escenario básico: 1) Modificar los datos pertinentes.
2) Actualizar BD.



Nombre: Consultar datos personales médico

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Médico consultado.

Objetivo: Consultar los datos personales de un médico.

Escenario básico: 1) Buscar al médico seleccionando el/los criterios de búsqueda.

2) Visualizar listado de médicos que corresponden con los criterios.

Opcional: Realizar de nuevo la búsqueda.

3) Seleccionar un médico.

4) Consultar datos del médico.

Nombre: Dar alta paciente

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Paciente dado de alta.

Objetivo: Dar de alta a un nuevo paciente del centro de salud.

Escenario básico: 1) Introducir datos personales del paciente.

2) Actualizar BD.

Opcional: Introducir alergias y/o asignar un médico.

Nombre: Modificar paciente

Actores: Administrador.

Precondición: Paciente dado de alta ó paciente consultado.

Poscondición: Datos paciente actualizados.

Objetivo: Modificar los datos de un paciente.

Escenario básico: 1) Modificar los datos pertinentes.

2) Actualizar BD.

Opcional: Añadir y/o modificar alergias. Asignar o cambiar de médico.

Nombre: Generar cita

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Paciente citado.

Objetivo: Citar a un paciente para acudir a la consulta.

Escenario básico: 1) Introducir datos necesarios.

2) Actualizar BD.

3) Visualizar cita.

Opcional: Imprimir cita.



Nombre: Dar alta medicamento

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Medicamento dado de alta.

Objetivo: Introducir nuevos medicamentos en la BD.

Escenario básico: 1) Introducir datos necesarios.
2) Actualizar BD.

Nombre: Modificar medicamento

Actores: Administrador.

Precondición: Medicamento consultado ó medicamento dado de alta.

Poscondición: Medicamento actualizado.

Objetivo: Modificar los datos de un medicamento.

Escenario básico: 1) Modificar los datos pertinentes.
2) Actualizar BD.

Nombre: Dar alta vacuna

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Nueva vacuna introducida.

Objetivo: Introducir una nueva vacuna en el calendario de vacunaciones.

Escenario básico: 1) Introducir datos vacuna.
2) Actualizar BD.

Nombre: Cambiar calendario de vacunaciones

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Calendario actualizado.

Objetivo: Modificar el calendario de vacunaciones.

Escenario básico: 1) Seleccionar el nombre de la vacuna que desea modificar.
Opcional: Seleccionar otra vacuna.
3) Pulsar sobre la vacuna.
4) Modificar datos de vacuna ó borrar vacuna.

Nombre: Cambiar mapa del centro

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Mapa actualizado.

Objetivo: Modificar el mapa del centro.

Escenario básico: **Opcional:** Visualizar imagen si existe.
1) Introducir nueva ruta o modificar la anterior.

2) Actualizar imagen.

Opcional: Visualizar imagen.



Nombre: Dar alta noticia

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Noticia dada de alta.

Objetivo: Introducir nuevas noticias en la Web.

Escenario básico: 1) Introducir datos pertinentes.
2) Validar noticia.

Nombre: Modificar noticia

Actores: Administrador.

Precondición: Noticia dada de alta ó noticia consultada.

Poscondición: Noticia actualizada.

Objetivo: Modificar una noticia.

Escenario básico: 1) Modificar los datos pertinentes.
2) Actualizar BD.

Nombre: Borrar noticia

Actores: Administrador.

Precondición: Noticia dada de alta ó noticia consultada.

Poscondición: Noticia borrada.

Objetivo: Borrar una noticia.

Escenario básico: 1) Pulsar sobre borrar.
2) Actualizar BD.

Nombre: Crear asunto

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Asunto creado.

Objetivo: Crear un asunto para el foro.

Escenario básico: 1) Introducir nombre.
2) Actualizar BD.

Nombre: Modificar asunto

Actores: Administrador.

Precondición: Asunto creado.

Poscondición: Asunto modificado.

Objetivo: Modificar el nombre de un asunto.

Escenario básico: 1) Modificar nombre.
2) Actualizar BD.



Nombre: Borrar asunto

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Asunto borrado.

Objetivo: Borrar un asunto creado en el foro.

Escenario básico: 1) Seleccionar el asunto a borrar.
2) Actualizar BD.

Nombre: Borrar pregunta

Actores: Administrador.

Precondición: Administrador identificado.

Poscondición: Pregunta borrada.

Objetivo: Borrar una pregunta creada en el foro.

Escenario básico: 1) Seleccionar asunto.
2) Seleccionar pregunta a borrar.
3) Actualizar BD.

4.2.- FUNCIONALIDAD

En este apartado, se explicarán detalladamente las distintas funcionalidades que forman parte de la aplicación Web.

Gracias a ello, evitaremos repetirnos a lo largo de los demás documentos, consiguiendo así generalizar en este apartado el uso de la aplicación.

- **Identificarse:** El usuario introducirá su DNI y contraseña, pulsando posteriormente el botón aceptar. Después se verificará su existencia en la BD. En caso de existir, se redirigirá a la pantalla principal del médico ó administrador según sea el caso; si el usuario no existe, se mostrará un error y se dará la posibilidad de volver a introducir los datos.
- **Consultar datos personales paciente:** El médico buscará a un paciente por número de historial. DNI, nombre y/o apellidos. Tras pulsar sobre el botón buscar, aparecerá un listado con las coincidencias encontradas o en caso contrario un mensaje avisándole que no existen coincidencias.
Tras esto, el médico podrá actuar de tres maneras:
 - Volver a realizar la búsqueda, procediendo de la misma manera.
 - Pulsar sobre uno de los resultados, en caso de haberse obtenido, apareciendo así los datos relativos a ese paciente.
 - Terminar sin hacer nada.



- **Modificar datos paciente:** El médico tras haber pulsado en un resultado de la lista obtenida en *consultar datos personales paciente*, podrá modificar cualquiera de los datos o alergias de un paciente, e incluso añadir una nueva alergia. Una vez pulse aceptar, se notificarán los cambios en la BD. En caso de producirse algún error, aparecerá un mensaje y se dará la oportunidad de volver a modificar los datos.
- **Generar diagnóstico:** El médico seleccionará el número de historial del paciente y pulsará sobre el botón generar diagnóstico. Después introducirá los datos necesarios para rellenar el diagnóstico. Una vez pulse guardar se comprobará si los datos son correctos, en caso de serlo se dará opción a recetar un medicamento, citar al paciente, realizar el seguimiento de Sintrom (si ha seleccionado la acción) ó por defecto no hacer nada más y terminar así. En caso de que se produzca algún error al generar el diagnóstico, aparecerá dicho error y se dará opción a resolver los errores. Una vez creado el diagnóstico el médico podrá modificar los datos si lo considera necesario.
- **Realizar seguimiento de Sintrom:** El médico introducirá los datos pertinentes y pulsará aceptar. Si todo es correcto aparecerá una pantalla con la DTS recomendada. Si la DTS fuera $<0,5\text{mg}$ ó $>30\text{mg}$ entonces se daría opción a que el médico introdujera la DTS manualmente, volviendo por tanto a validar. Finalmente se podrá pulsar el botón de citar paciente ó recetar un medicamento. Si se produce algún error, se informará de ello dando de nuevo la oportunidad de corregirlo.
- **Recetar medicamento:** Si el médico no llega a esta opción desde un diagnóstico o un seguimiento de Sintrom, deberá seleccionar el número de historia del paciente al que desea extender la receta, después y para todos los casos el médico buscará el medicamento que desea recetar introduciendo el nombre y/o tipo. Cuando pulse el botón buscar, aparecerá un listado con las coincidencias y en caso de no existir coincidencias, saldrá un mensaje avisándole de que no existen coincidencias.
Tras esto el médico podrá:
 - Volver a realizar la búsqueda, procediendo de la misma forma.
 - Pulsar uno de los elementos de la lista, de forma, que aparezca generada la receta. Una vez generada la receta el médico podrá modificar manualmente la posología, volviendo a guardar la receta. Finalmente podrá pulsar el botón imprimir para que la receta pueda ser dada al paciente y/o sobre volver para recetar otro medicamento.
- **Citar paciente:** Si el médico no llega a esta opción desde un diagnóstico o un seguimiento de Sintrom, deberá seleccionar el número de historia del paciente al que desea citar, después y para todos los casos el médico seleccionará la fecha y hora de la cita y pulsará guardar, en caso de no ser correctos se le avisará del error. Finalmente se visualizará la cita pudiendo imprimirla y/o pulsar sobre volver para citar otra vez al paciente.



- **Consultar historial:** El médico buscará el historial de un paciente por número de historial. DNI, nombre y/o apellidos. Tras pulsar sobre el botón buscar, aparecerá un listado con las coincidencias encontradas o en caso contrario un aviso de que no existen coincidencias. Después el médico, podrá volver a realizar la búsqueda, salir sin hacer nada ó pulsar sobre un paciente de forma que aparecerá un listado con todas los diagnósticos que tiene su historial dándole la oportunidad de consultar una de ellos pulsando en él. Si el diagnóstico contiene seguimiento de Sintrom, se podrán visualizar los resultados de dicho seguimiento.
- **Consultar medicamento:** El médico ó administrador buscará el medicamento por nombre y/o tipo. Cuando pulse el botón buscar, aparecerá un listado con las coincidencias y en caso de no existir coincidencias, saldrá un mensaje avisándole de que no existen coincidencia
Tras esto el médico podrá:
 - Volver a realizar la búsqueda, procediendo de la misma forma.
 - Pulsar uno de los elementos de la lista y consultar todos sus datos.
 - Salir sin realizar ninguna acción.
- **Consultar agenda:** El médico introducirá la fecha de la cual quiere consultar sus citas, una vez pulse sobre el botón consultar aparecerá un listado con todas las citas que tiene para esa fecha. En caso de no tener ninguna cita, la aplicación le mostrará un aviso de ello. Si lo desea, el médico tendrá opción a consultar otra fecha.
- **Consultar foro:** El médico ó el usuario seleccionará uno de los asuntos que existen creados en el foro, una vez seleccionado aparecerán todas las preguntas pertenecientes a ese asunto. Si lo desea podrá pulsar sobre una pregunta para ver sus respuestas asociadas.
- **Responder al foro:** Una vez que el médico ha pulsado sobre una pregunta, tendrá opción a responderla pulsando sobre el botón responder. Una vez respondida se actualizará el foro para que aparezca su respuesta.
- **Preguntar en el foro:** Cualquier usuario tras realizar una consulta del foro, tiene la opción de pulsar sobre el botón preguntar. Una vez pulsado, deberá introducir su nombre ó alias y formular la pregunta, tras pulsar sobre el botón aceptar, se comprobará que no existe ningún error. En caso de estar todo correcto, se actualizará el foro para que aparezca su pregunta reflejada, sino, aparecerá un mensaje informándole del error y dándole opción a solucionarlo.
- **Visualizar directorio:** Cualquier usuario podrá consultar los datos de los médicos del centro, pulsando sobre esta opción y seleccionando cualquier médico del centro.



- **Consultar noticia:** Cualquier usuario podrá seleccionar el título de una noticia y visualizar su contenido.
- **Consultar calendario de vacunaciones:** Cualquier usuario podrá consultar los datos pertinentes a las vacunas, pulsando sobre esta opción.
- **Visualizar mapa del centro:** Cualquier usuario podrá consultar el mapa del centro de salud, pulsando sobre esta opción, con el fin de saber la localización de todos los despachos de los médicos.
- **Dar alta administrador:** Un administrador, podrá dar de alta a otros administradores introduciendo sus datos personales. Tras ello, deberá pulsar sobre el botón aceptar, almacenándose el nuevo registro en la BD. Si se produce algún error se informará de ello, de forma que pueda solucionarlo, sino aparecerá la misma pantalla con los datos introducidos ofreciéndole la posibilidad de modificar alguno de los datos si no es correcto.
- **Modificar perfil:** Un administrador podrá modificarse a sí mismo cambiando cualquiera de los datos almacenados y pulsando sobre el botón aceptar, para guardar los cambios. Como siempre se comprobará si existe algún error antes de actualizar la BD, en caso de ser así se notificará al usuario para que pueda corregirlo.
- **Dar alta médico:** Cualquier administrador podrá dar de alta a un médico introduciendo sus datos personales y pulsando sobre el botón aceptar para almacenar el nuevo registro. Esta funcionalidad funciona de la misma manera que "dar alta administrador".
- **Consultar médico:** Cualquier administrador puede consultar los datos personales de un médico, para ello, deberá buscar al médico introduciendo su número de colegiado, su DNI, su nombre y/o sus apellidos. Tras pulsar sobre el botón buscar, aparecerá un listado con las coincidencias encontradas o en caso contrario un mensaje avisándole de que no existen coincidencias. Tras esto el administrador podrá:
 - Volver a realizar la búsqueda, procediendo de la misma forma.
 - Pulsar uno de los elementos de la lista y consultar así los datos del médico.
 - Salir sin realizar ninguna acción.
- **Modificar médico:** Cualquier administrador tras haber pulsado en un resultado de la lista obtenida en *consultar médico* ó tras dar de alta a un administrador, podrá modificar cualquiera de los datos personales del médico. Una vez pulse aceptar, se notificarán los cambios en la BD. En caso de producirse algún error, aparecerá un mensaje y se dará la oportunidad de volver a modificar los datos.



- ***Dar alta paciente:*** Cualquier administrador podrá dar de alta a un paciente introduciendo sus datos personales, alergias y asignándole un médico. Tras pulsar sobre el botón aceptar se almacenará el nuevo registro. Esta funcionalidad funciona de la misma manera que “*dar alta administrador*”, a excepción de que si tiene que introducir una alergia, el administrador deberá pulsar sobre el botón añadir alergia apareciendo por lo tanto una nueva pantalla donde podrá introducir los datos de la alergia, una vez pulse aceptar, se redireccionará a la página de los datos personales reflejándose la nueva alergia introducida. De la misma manera, si desea asignar un médico deberá pulsar sobre el botón asignar médico apareciendo por lo tanto una nueva pantalla donde podrá seleccionar un médico del centro, una vez pulse aceptar, se redireccionará a la página de los datos personales reflejándose dicha asignación y dando la oportunidad de cambiar de médico.
- ***Generar cita:*** Cualquier administrador podrá generar citas, estas citas por lo general se corresponderán con la primera cita que tiene un paciente con su médico, ya que el resto las dará el propio médico.
El administrador deberá seleccionar el paciente a citar (se obvia poner el médico ya que debe estar asignado), una vez seleccionados elegirá la fecha y hora, cita y pulsará guardar, en caso de no ser correctos se le avisará del error. Finalmente se visualizará la cita pudiendo imprimirla y/o pulsar sobre volver para citar mas pacientes.
- ***Dar alta medicamento:*** Cualquier administrador podrá dar de alta nuevos medicamentos introduciendo los datos necesarios. Tras pulsar sobre el botón aceptar se almacenará el nuevo registro. Esta funcionalidad funciona de la misma manera que “*dar alta administrador*”.
- ***Dar alta vacuna:*** Cualquier administrador podrá dar de alta una nueva vacuna introduciendo los datos necesarios. Tras pulsar sobre el botón aceptar se almacenará el nuevo registro. Esta funcionalidad funciona de la misma manera que “*dar alta administrador*”.
- ***Cambiar calendario de vacunaciones:*** Cualquier administrador podrá consultar, modificar y/o borrar una vacuna de la BD, para ello, el administrador seleccionará el nombre de la vacuna que desea buscar. Tras pulsar sobre el botón mostrar, aparecerá un listado con las coincidencias encontradas o en caso contrario un mensaje de error.
Tras esto el administrador podrá:
 - Volver a realizar seleccionar otra vacuna, procediendo de la misma forma.
 - Pulsar uno de los elementos de la lista y consultar así los datos de la vacuna. En esta pantalla se dará opción a modificar la vacuna, si pulsa sobre el botón guardar, o a borrarla de la BD pulsando el botón borrar.
 - Salir sin realizar ninguna acción.

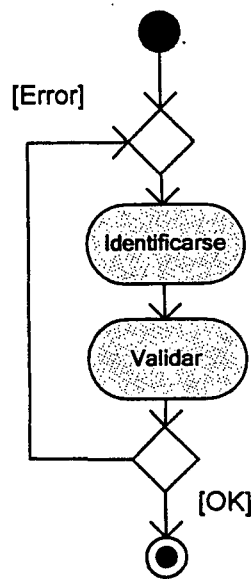


- **Cambiar mapa centro:** Cualquier administrador podrá cambiar la imagen del mapa del centro que se muestra en la aplicación, para ello, deberá introducir la nueva ruta donde se encuentra el documento con extensión .jpg del nuevo mapa.
Finalmente si lo desea puede pulsar sobre el botón visualizar para ver la imagen.
- **Dar alta noticia:** Cualquier administrador podrá dar de alta una nueva noticia introduciendo los datos necesarios. Tras pulsar sobre el botón aceptar se almacenará el nuevo registro. Esta funcionalidad funciona de la misma manera que “*dar alta administrador*”, además añade la posibilidad de borrar la noticia pulsando sobre el botón borrar.
- **Crear asunto:** Cualquier administrador podrá crear un nuevo asunto en el foro introduciendo el título. Tras pulsar sobre el botón aceptar se almacenará el nuevo asunto.
- **Borrar asunto:** Cualquier administrador podrá borrar cualquier asunto creado en el foro, desapareciendo de esta forma todas las preguntas y respuesta que este contiene. Para ello deberá seleccionar el asunto a borrar y pulsar sobre el botón borrar.
- **Borrar pregunta:** Cualquier administrador podrá borrar cualquier pregunta creada en el foro, desapareciendo también todas las respuestas asociadas a esa pregunta en caso de existir alguna respuesta. Para ello deberá seleccionar un asunto y una pregunta, y tras ello pulsar sobre el botón borrar.

4.3.- DIAGRAMAS DE ACTIVIDAD

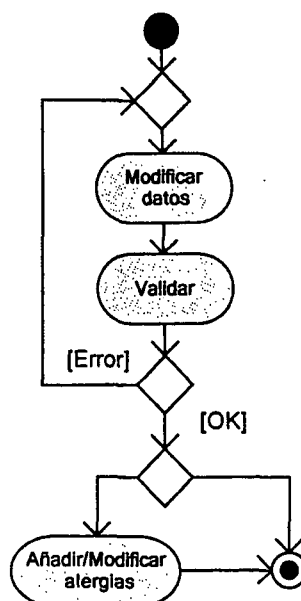
Mediante los diagramas de actividad se mostrará una descripción gráfica en la que se especificará, más detalladamente lo que ocurre dentro de cada caso de uso.

- *Identificase*



- Figura 21: Diagrama actividad identificarse -

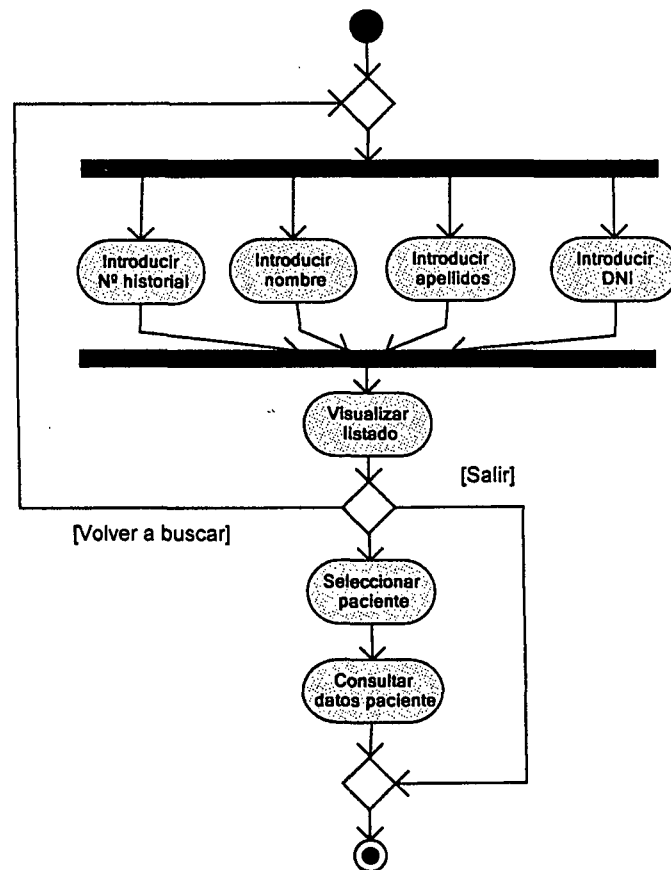
- *Modificar datos paciente*



- Figura 22: Diagrama actividad modificar datos paciente -

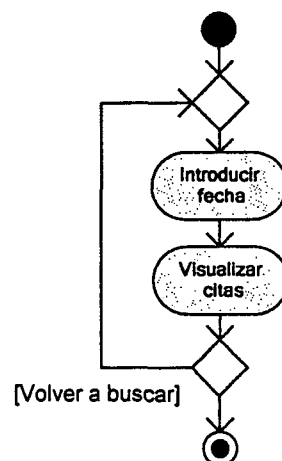


- *Consultar datos personales paciente*



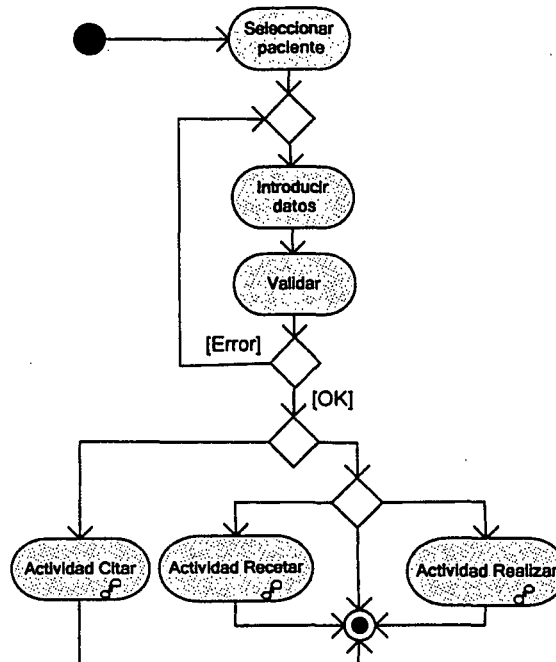
- Figura 23: Diagrama actividad consultar datos personales paciente -

- *Consultar agenda*



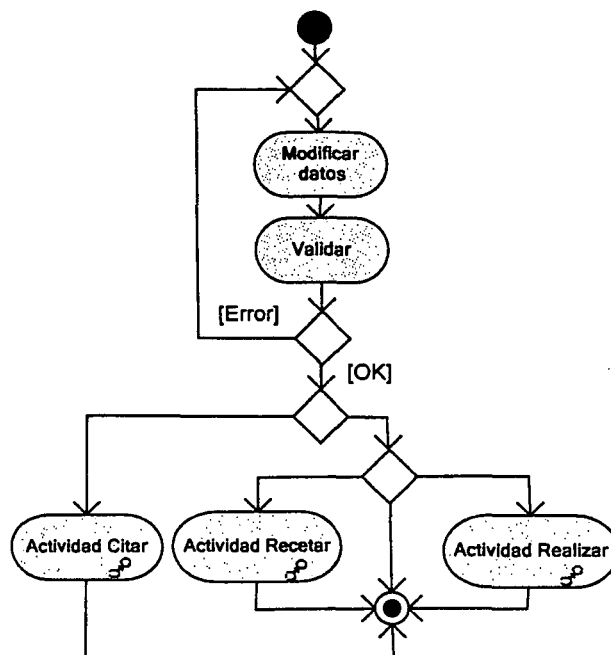
- Figura 24: Diagrama actividad consultar agenda -

- **Generar diagnóstico:** Al final de este diagrama se añaden tres subactividades citar paciente, recetar medicamento y realizar seguimiento de Sintrom, ya que aunque son tres opciones que se pueden realizar en esta funcionalidad, son actividades a parte, que tienen su propio diagrama de actividad.



- Figura 25: Diagrama actividad generar diagnóstico -

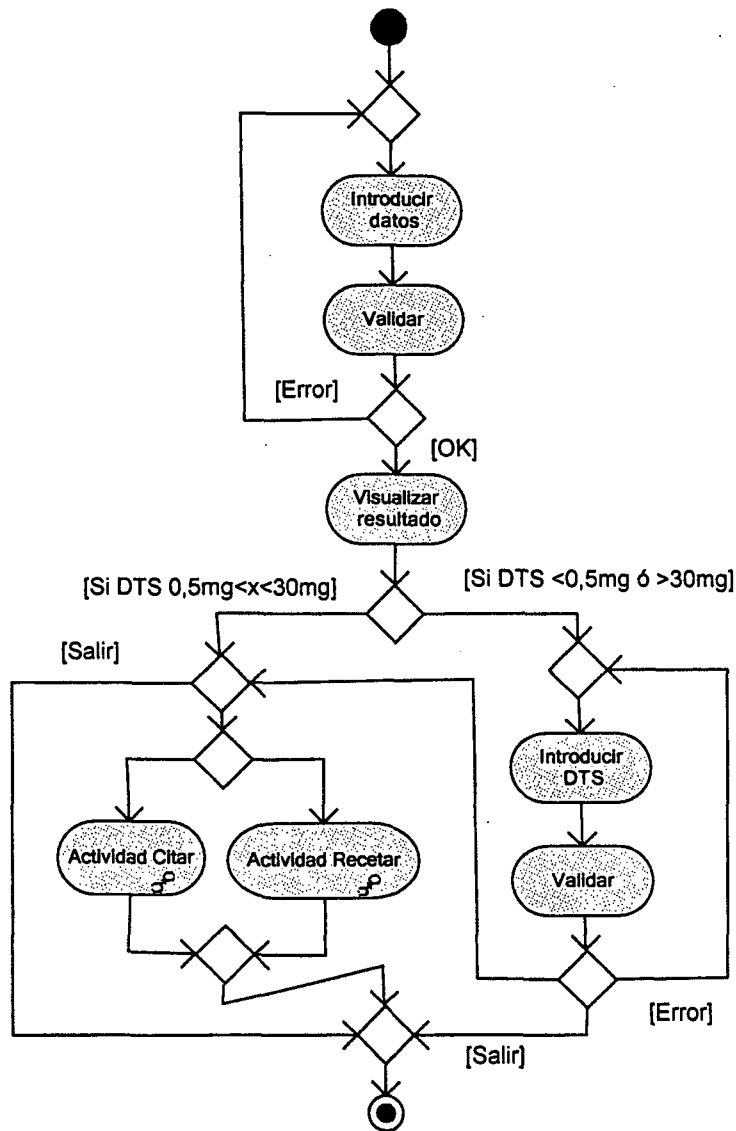
- **Modificar diagnóstico:**



- Figura 26: Diagrama actividad generar diagnóstico -

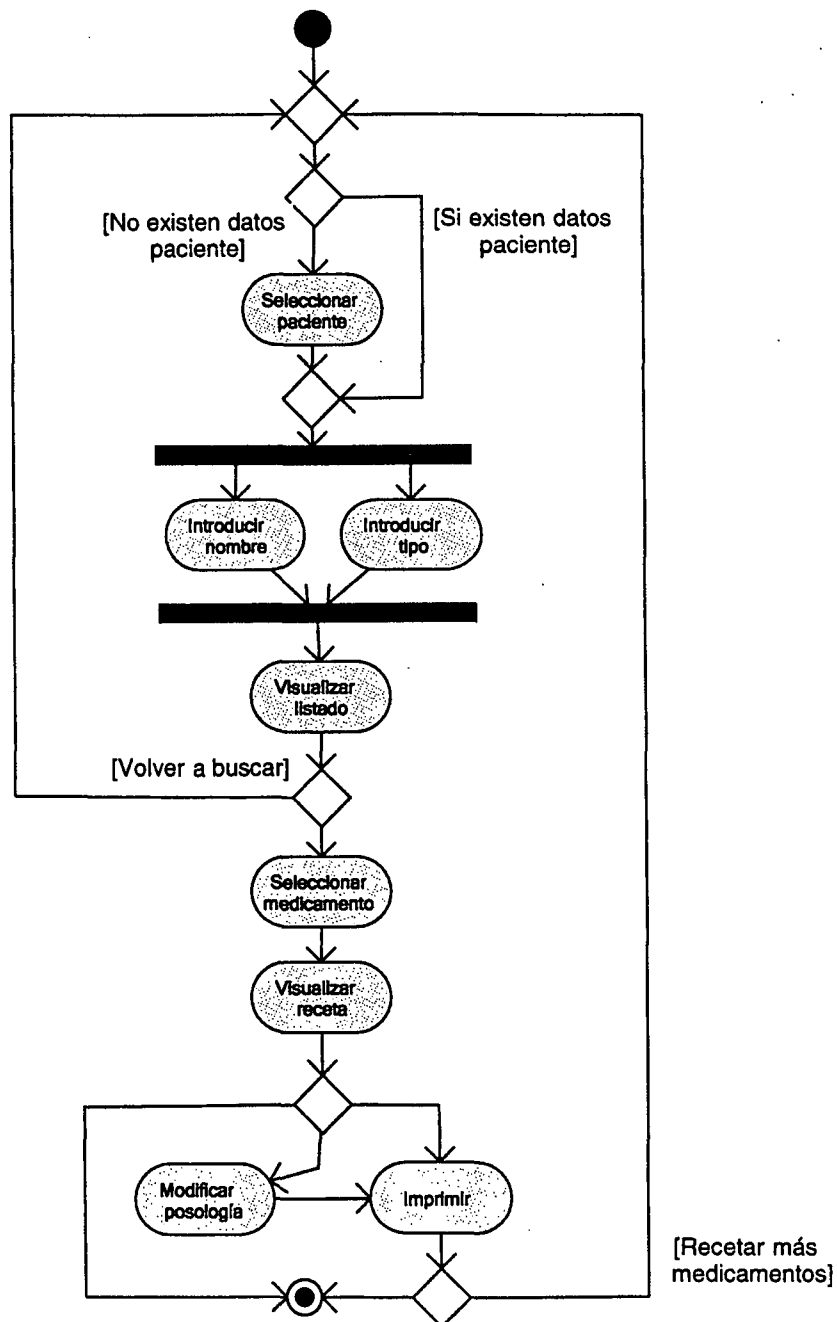


- **Realizar seguimiento de Sintrom:** Se han añadido como subactividades citar paciente y recetar medicamento, ya que ambas tienen su propio diagrama de actividad.



- Figura 27: Diagrama actividad realizar seguimiento de Sintrom -

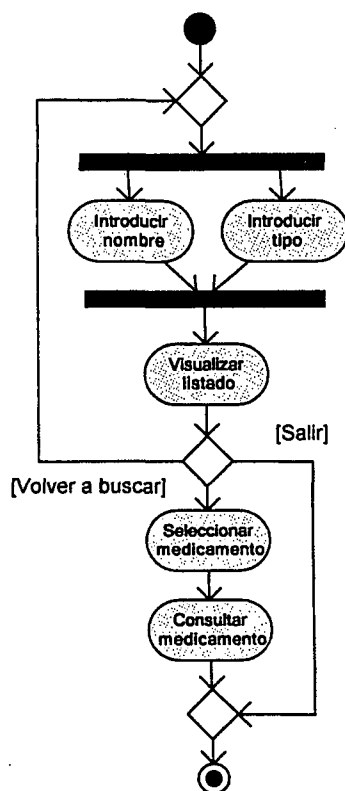
- **Recetar medicamento**



- Figura 28: Diagrama actividad recetar medicamento -

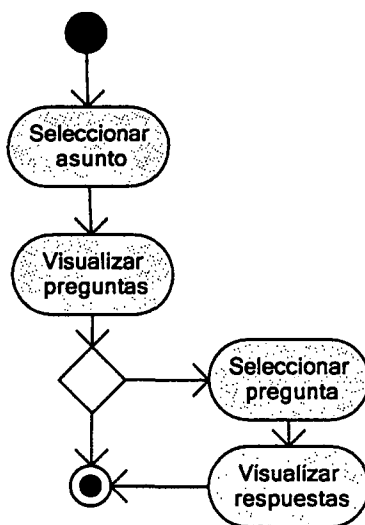


- **Consultar medicamento**



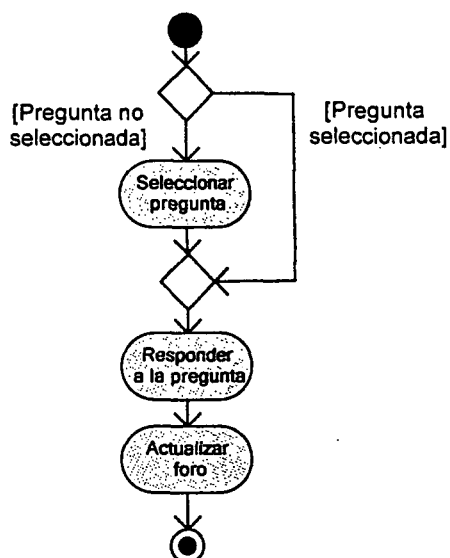
- Figura 29: Diagrama actividad consultar medicamento -

- **Consultar foro**



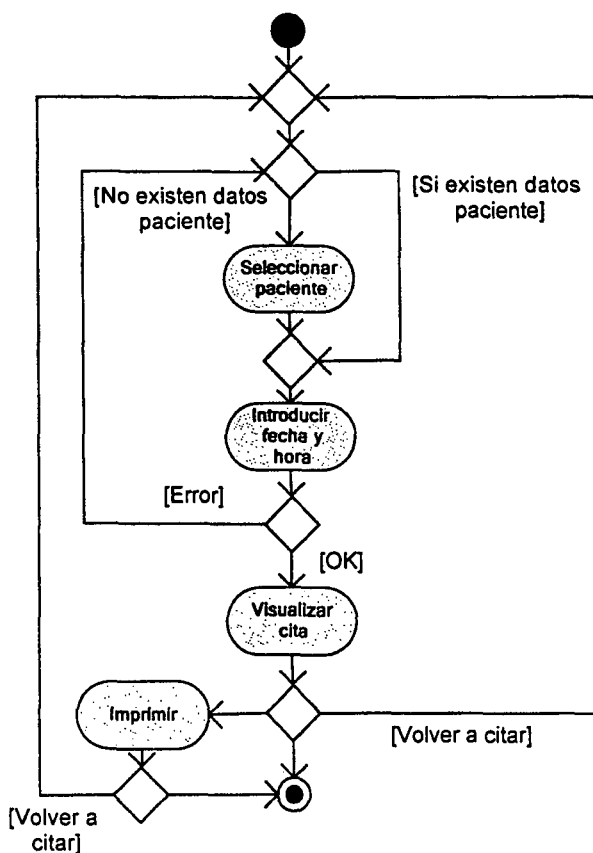
- Figura 30: Diagrama actividad consultar foro -

- *Responder al foro*



- Figura 31: Diagrama actividad responder al foro -

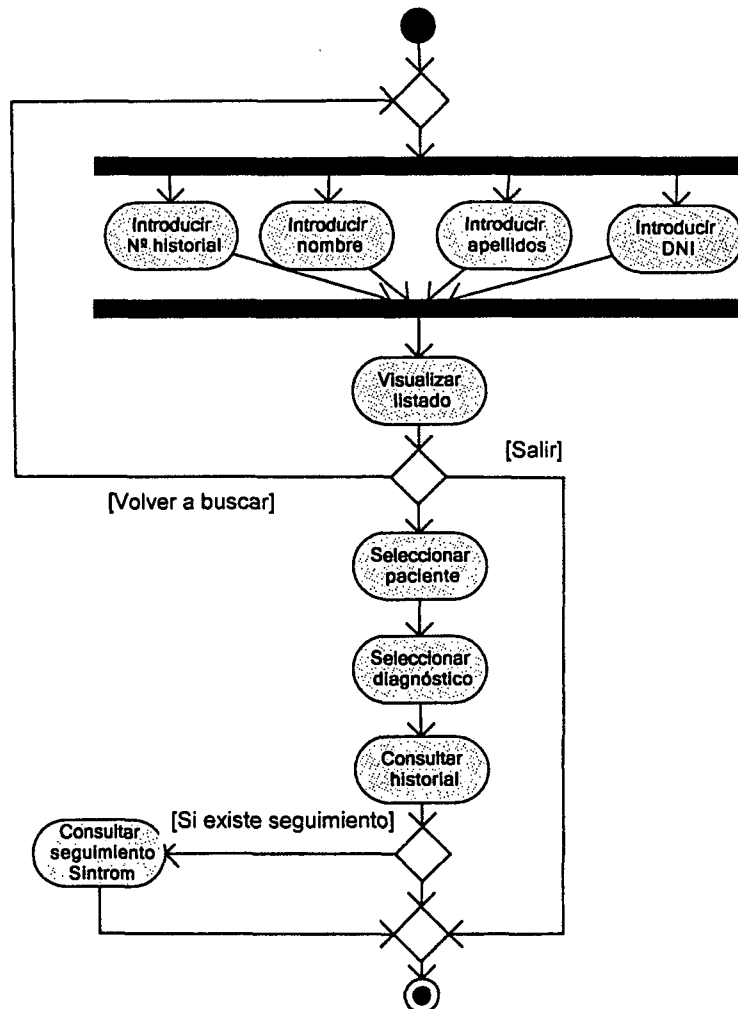
- *Citar paciente*



- Figura 32: Diagrama actividad citar paciente -



- **Consultar historial**

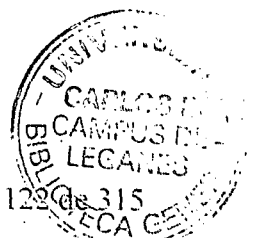


- Figura 33: Diagrama actividad consultar historial -

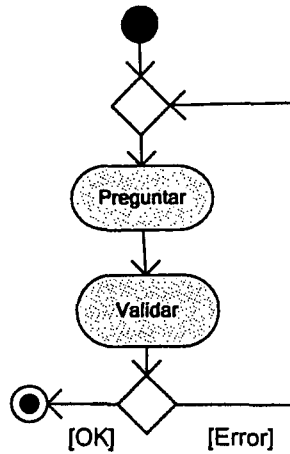
- **Consultar calendario de vacunaciones**



- Figura 34: Diagrama actividad consultar calendario de vacunaciones -



- *Preguntar en el foro*



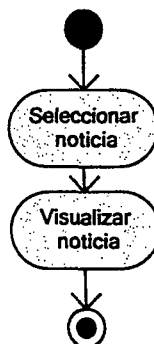
- Figura 35: Diagrama actividad preguntar en el foro -

- *Visualizar directorio*



- Figura 36: Diagrama actividad visualizar directorio -

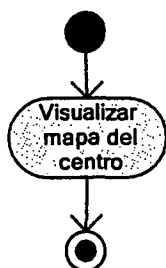
- *Consultar noticia*



- Figura 37: Diagrama actividad consultar noticia -

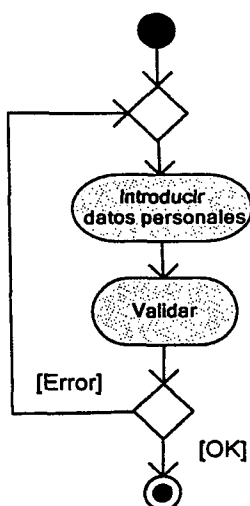


- *Visualizar mapa del centro*



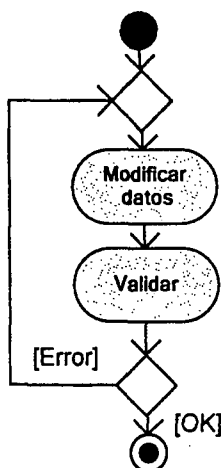
- Figura 38: Diagrama actividad visualizar mapa del centro -

- *Dar alta administrador*



- Figura 39: Diagrama actividad dar alta administrador -

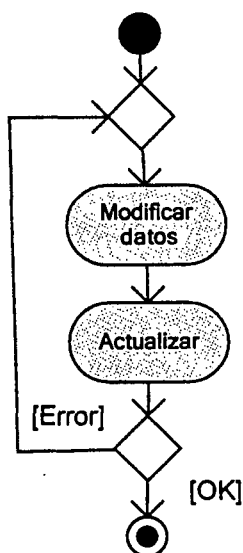
- *Modificar administrador*



- Figura 40: Diagrama actividad dar alta administrador -

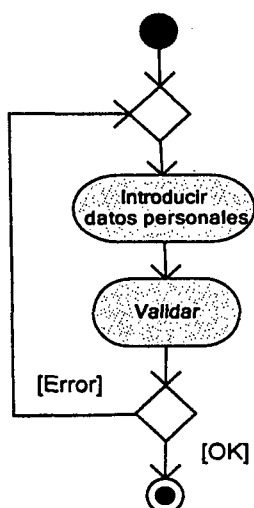


- *Modificar perfil*



- Figura 41: Diagrama actividad modificar perfil -

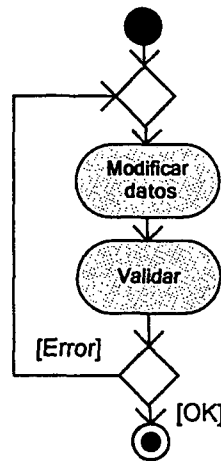
- *Dar alta médico*



- Figura 42: Diagrama actividad dar alta médico -

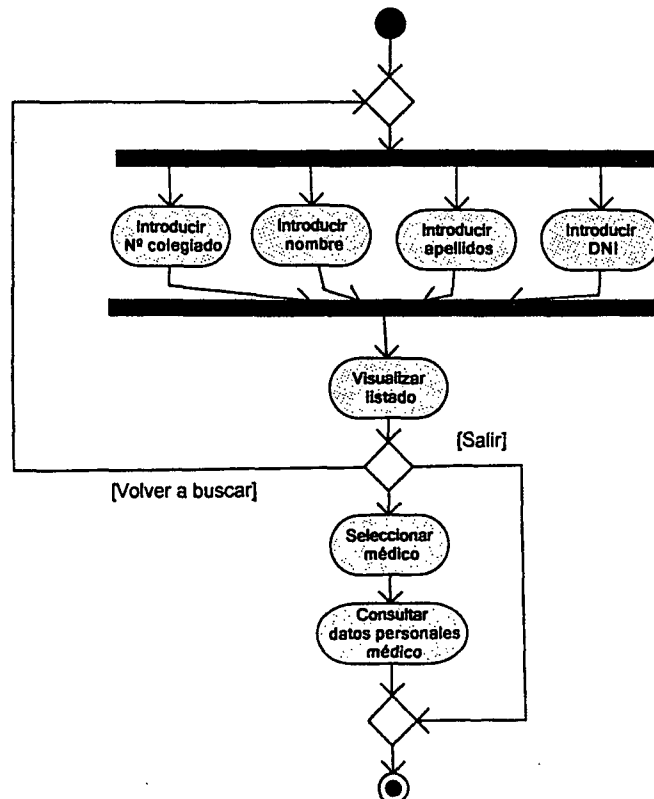


- *Modificar datos médico*



- Figura 43: Diagrama actividad dar alta médico -

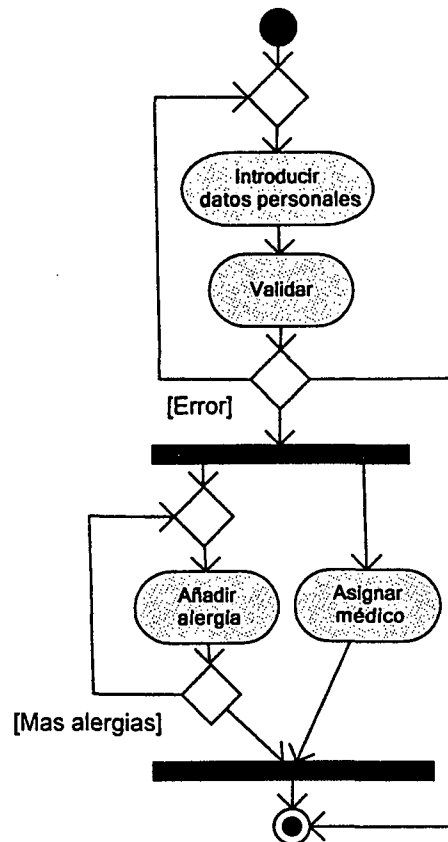
- *Consultar médico*



- Figura 44: Diagrama actividad consultar médico -

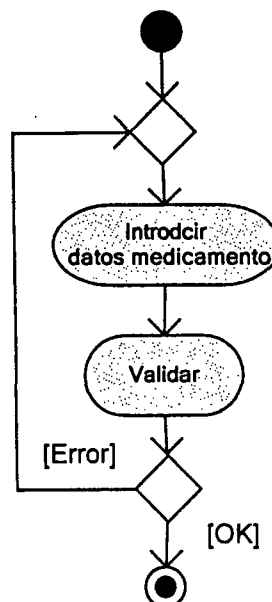


- *Dar alta paciente*



- Figura 45: Diagrama actividad dar alta paciente -

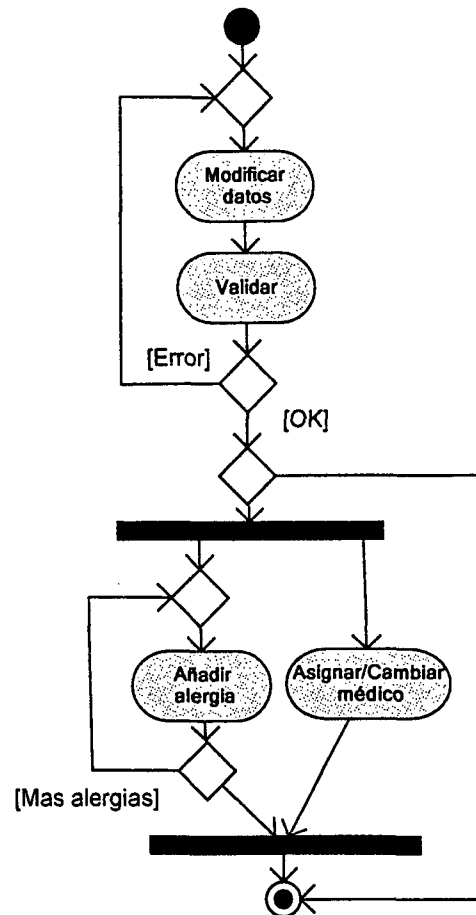
- *Dar alta medicamento*



- Figura 46: Diagrama actividad dar alta medicamento -

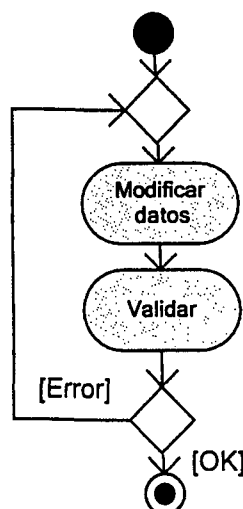


- *Modificar paciente*



- Figura 47: Diagrama actividad dar alta paciente -

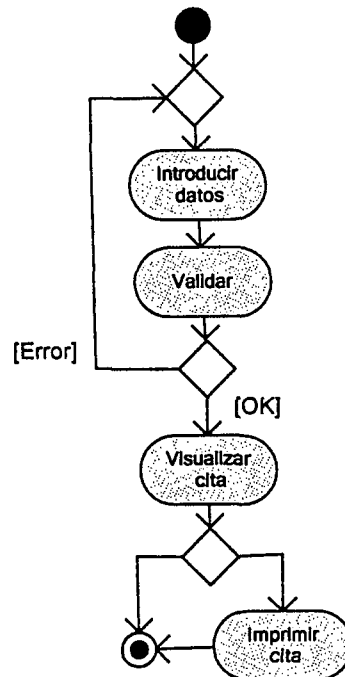
- *Modificar medicamento*



- Figura 48: Diagrama actividad dar alta medicamento -

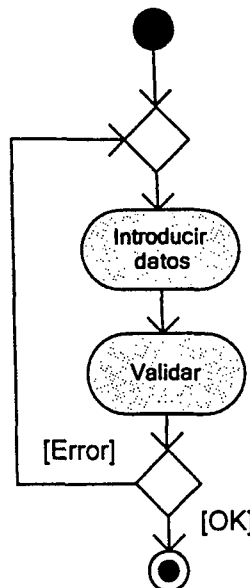


- **Generar cita**



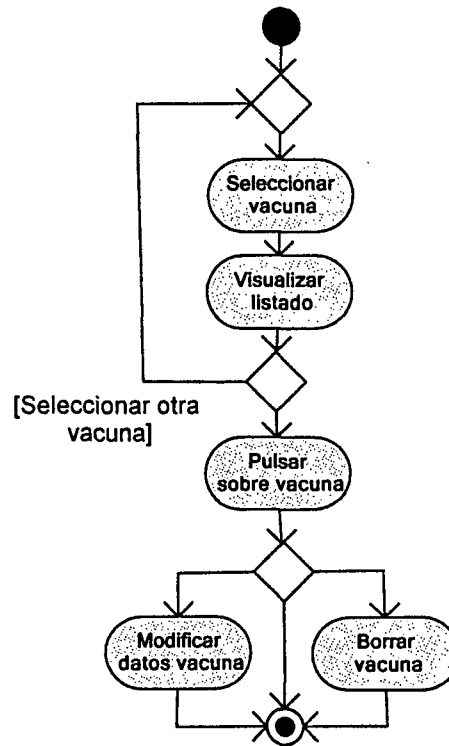
- Figura 49: Diagrama actividad generar cita -

- **Dar alta vacuna**



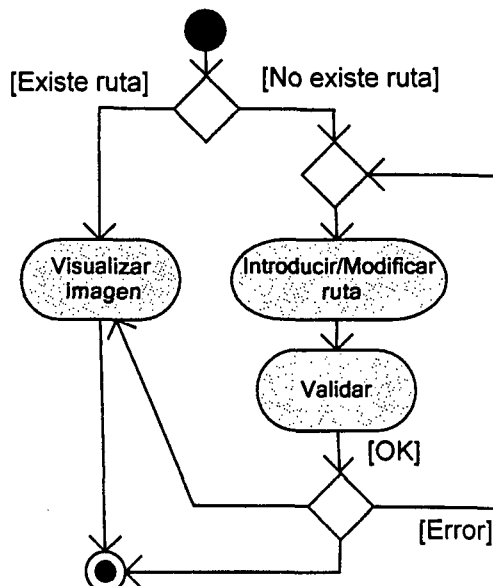
- Figura 50: Diagrama actividad dar alta vacuna -

- *Cambiar calendario de vacunaciones*



- Figura 51: Diagrama actividad cambiar calendario vacunaciones -

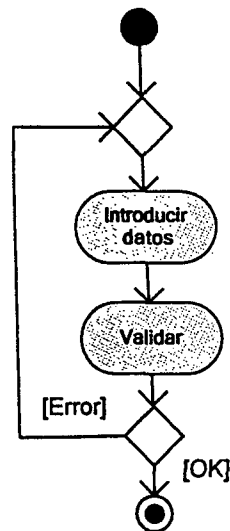
- *Cambiar mapa del centro*



- Figura 52: Diagrama actividad cambiar mapa del centro -

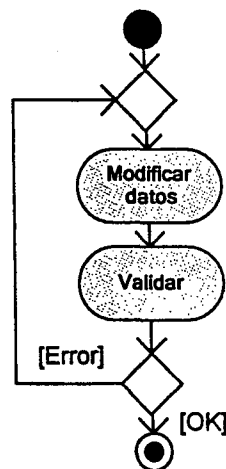


- ***Dar alta noticia***



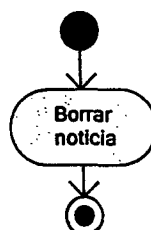
- Figura 53: Diagrama actividad dar alta noticia -

- ***Modificar noticia***



- Figura 54: Diagrama actividad dar alta noticia -

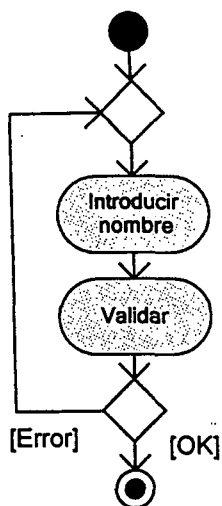
- ***Borrar noticia***



- Figura 55: Diagrama actividad borrar noticia -

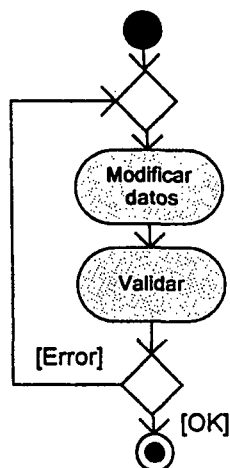


- **Crear asunto**



- Figura 56: Diagrama actividad crear asunto -

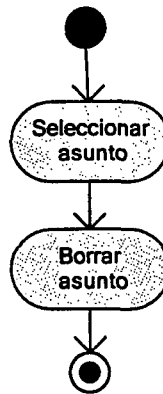
- **Modificar asunto**



- Figura 57: Diagrama actividad crear asunto -

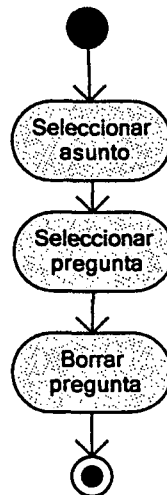


- **Borrar asunto**



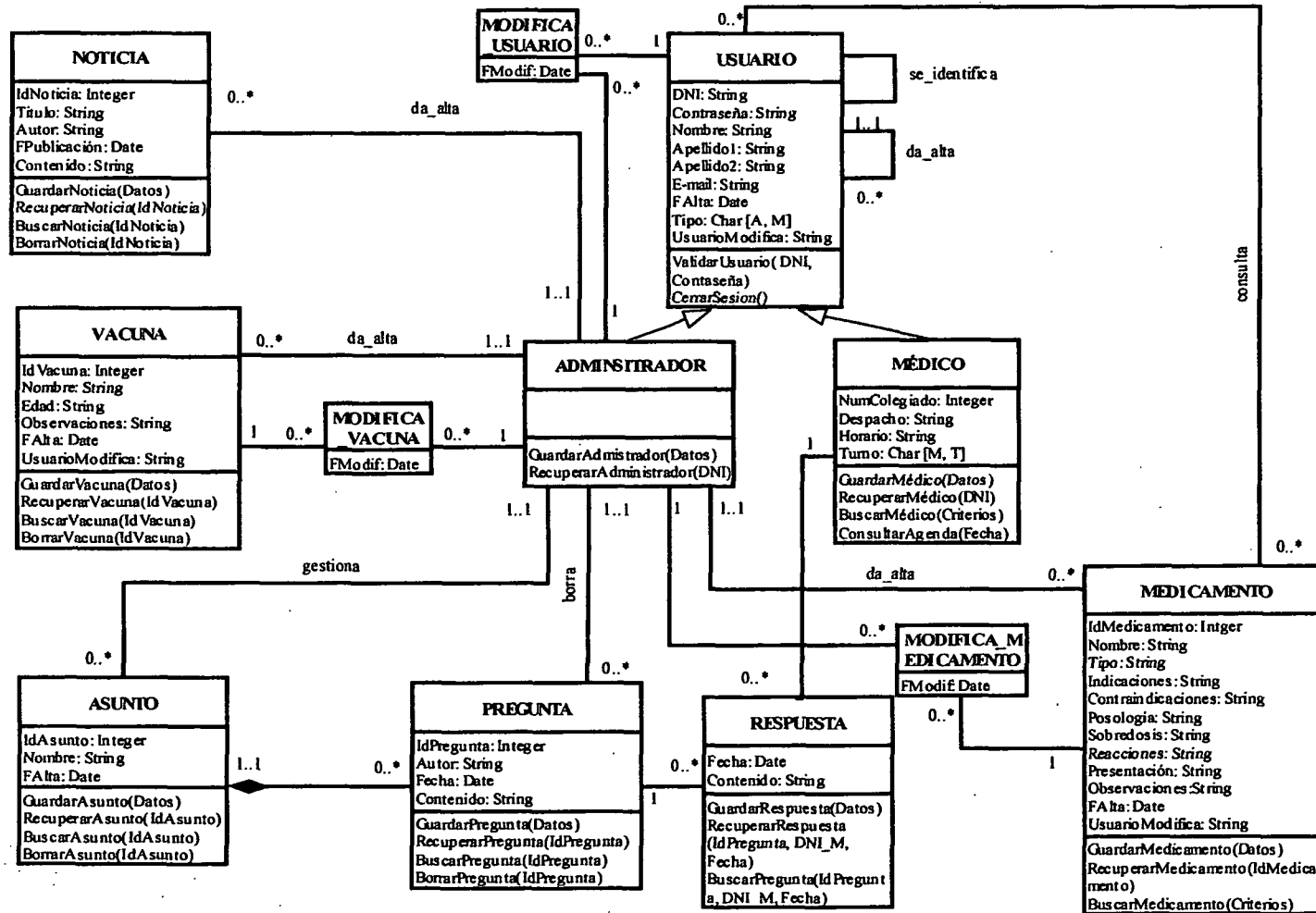
- Figura 58: Diagrama actividad borrar asunto -

- **Borrar pregunta**

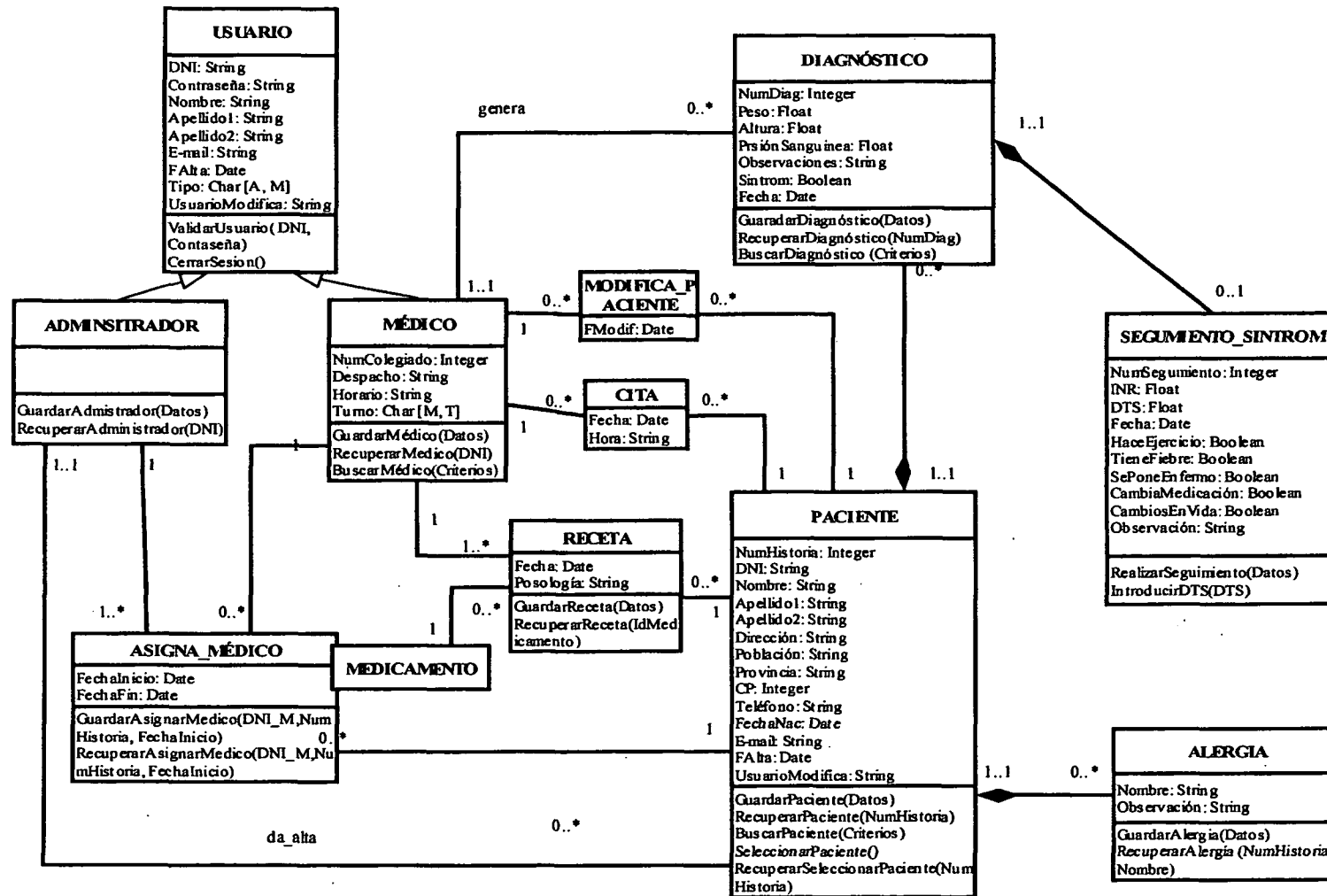


- Figura 59: Diagrama actividad borrar pregunta

4.4.- DIAGRAMA DE CLASES



- Figura 60: Diagrama de clases I -



- Figura 61: Diagrama de clases II -



Mediante este tipo de diagrama, se modela la estructura estática del sistema, recogiendo además los atributos y métodos que después se utilizarán para programar la aplicación Web.

4.4.1.- COMENTARIOS

- Las clases *modifica_vacuna*, *modifica_medimento*, *modifica_usuario*, *cita*, *modifica_paciente*, *respuesta*, *receta* y *asigna* son clases intermedias, ya que se permiten que existan tuplas repetidas, por ejemplo, una vacuna puede ser modificada en distintas fechas, de ahí que se produzca una clase intermedia.
- La clase *genera* es una clase-asociación entre diagnóstico y médico, ya que no se permiten tuplas repetidas, es decir, que un médico solo generará un diagnóstico una sola vez.
- Existe una jerarquía que surge de usuario y deriva en médico y administrador, de tal forma que se produce una herencia de atributos y métodos.
- La relación entre *asunto* y *pregunta*, *paciente* y *diagnóstico*, y *diagnóstico* y *seguimiento_sintrom*, y *paciente* y *alergia*, es una relación de composición, ya que existe una dependencia total entre ambas partes y por tanto las operaciones de borrado y modificación se propagaran del todo a las partes.

4.5.- DIAGRAMAS DE SECUENCIA

Los diagramas de secuencia son uno de los tipos de diagramas de interacción que existen.

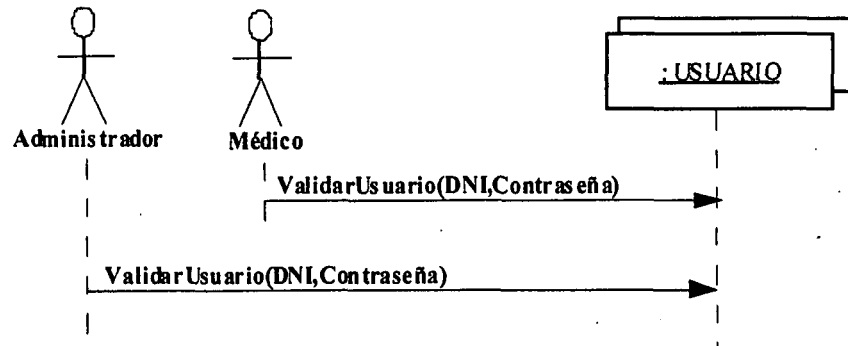
Estos diagramas modelan la vista dinámica del sistema añadiendo un nuevo elemento, el tiempo, describiendo además cómo colaboran o interactúan un grupo de objetos para llevar a cabo una funcionalidad del sistema. Especialmente se centran en reflejar la secuencia de mensajes que se envían en el tiempo.

Para representar estos diagramas se han utilizado mensajes síncronos. Además se ha representado en algunos casos como en "*identificarse*" la forma en la que se recorren todas las tuplas de la BD a través de la doble caja de la clase usuario.

Todos los métodos usados están reflejados en el diagrama de clases. Se aclara que cualquier modificación utiliza el método "*guardar...*", ya que este implica tanto guardar un nuevo elemento en la BD, como actualizarlo.

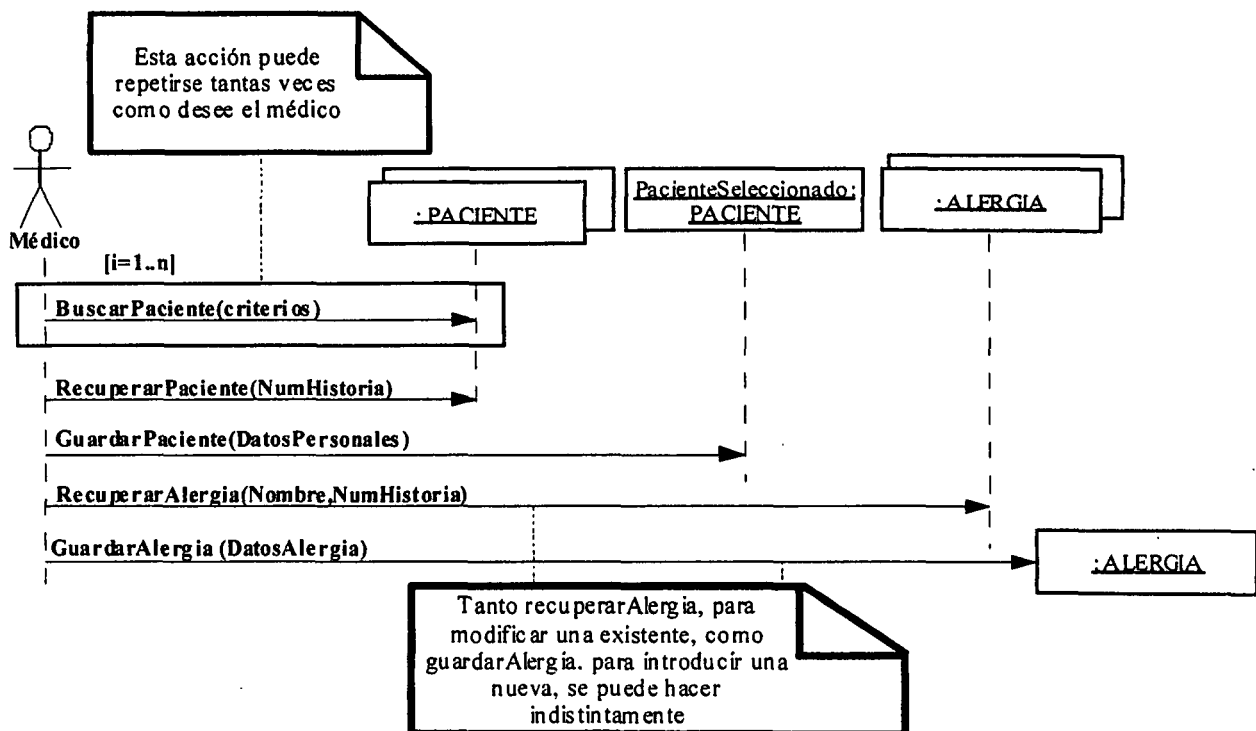
El hecho de que algunas funcionalidades no estén representadas, o se encuentren incompletas, se debe a que no se necesitan métodos específicos en Java para ellas, y otras son extremadamente sencillas por lo que no es necesario crear el diagrama de secuencia.

- **Identificarse**



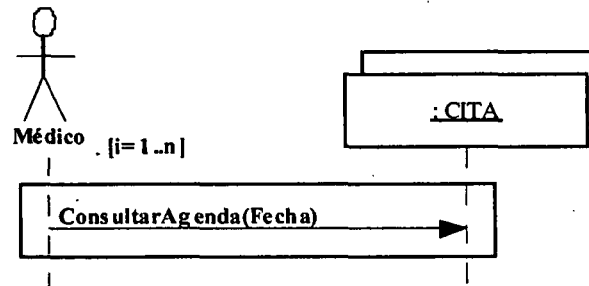
- Figura 62: Diagrama de secuencia identificarse -

- **Consultar datos personales paciente y modificar datos paciente:** Modificar paciente (*guardarPaciente*), y añadir o modificar alergias es algo opcional, si lo desea el médico únicamente puede consultar los datos del paciente y terminar sin hacer nada más.



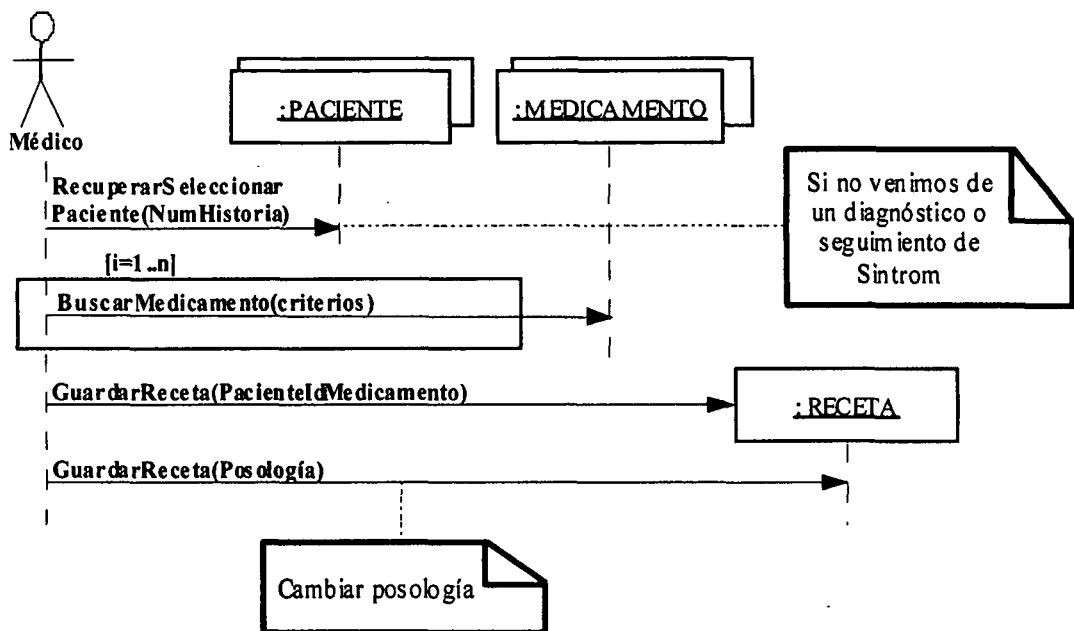
- Figura 63: Diagrama de secuencia consultar datos personales y modificar datos paciente -

- **Consultar agenda:** Tras llevar a cabo esta secuencia se mostraran por pantalla todas las citas que tiene un médico en la fecha introducida.



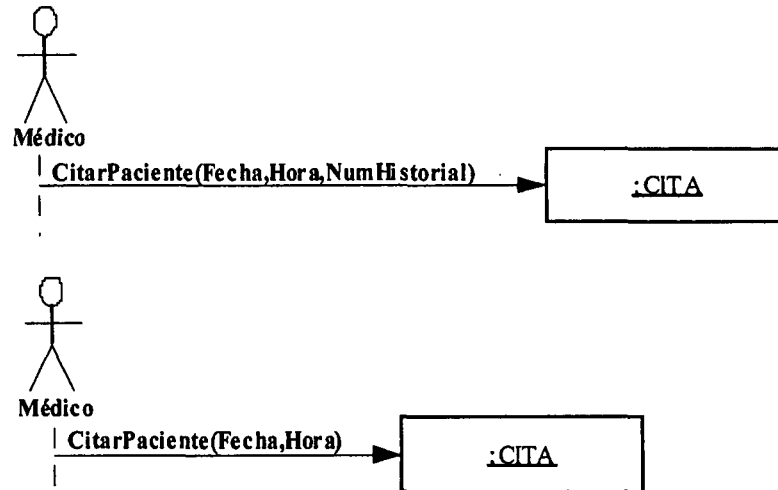
- Figura 64: Diagrama de secuencia consultar agenda -

- **Recetar medicamento:** La impresión de la receta no se ha reflejado porque es una acción JavaScript que se realiza contra el navegador.



- Figura 65: Diagrama de secuencia recetar medicamento -

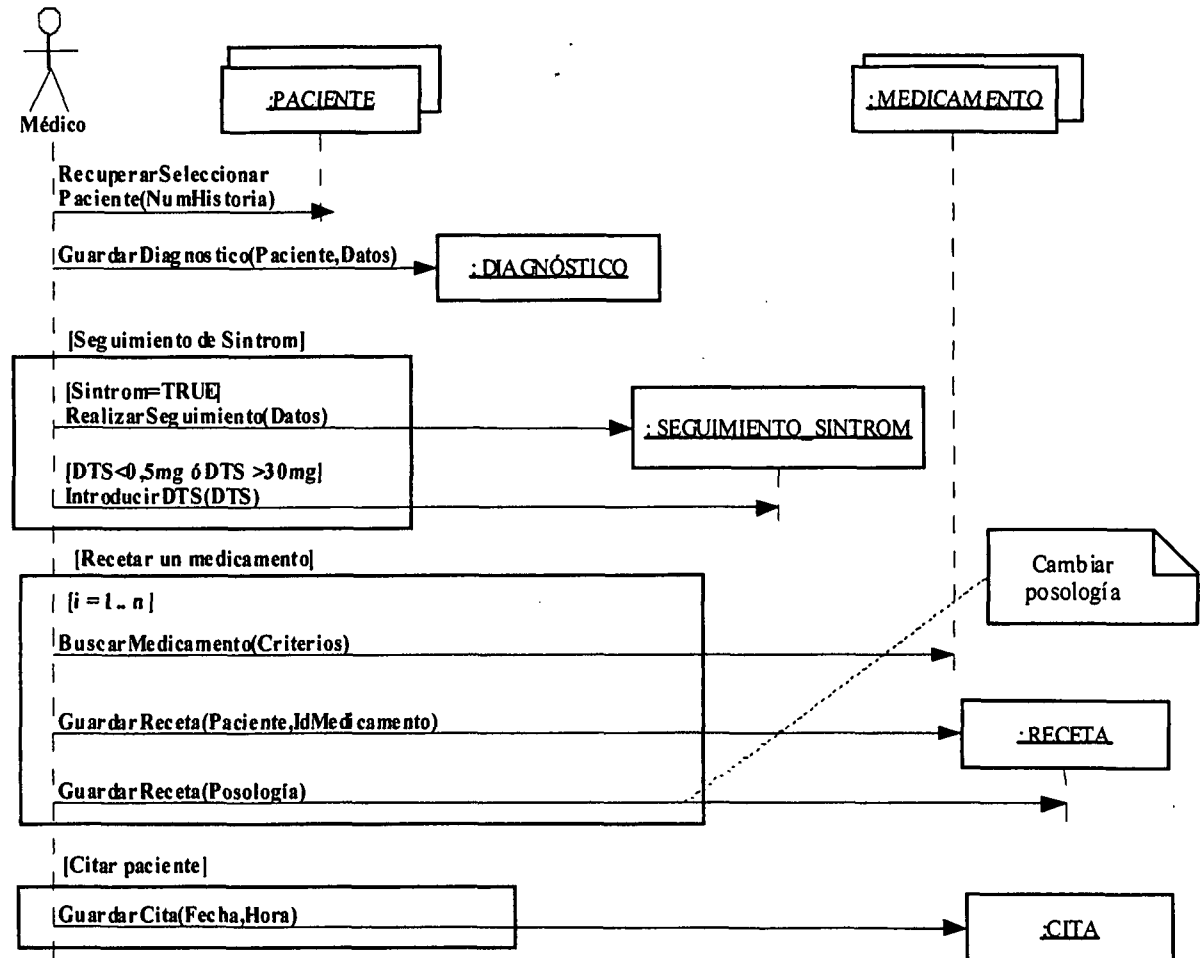
- **Citar paciente:** Se muestran dos diagramas para citar paciente, el primero de ellos refleja la secuencia que se producirá cuando un médico cita a un paciente, mientras que la segunda secuencia se dará cuando se cite a un paciente después de haber realizado un diagnóstico ó seguimiento de Sintrom.



- Figura 66: Diagrama de secuencia citar paciente -



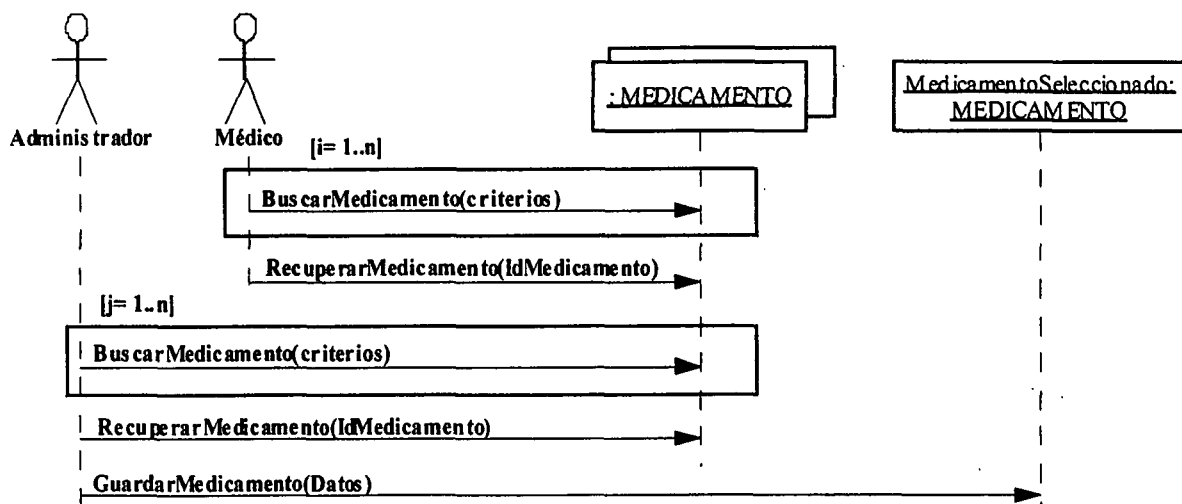
- **Generar diagnóstico y seguimiento de Sintrom:** No es necesario que tras generar un diagnóstico se realice un seguimiento de Sintrom, ni una receta, ni una cita, es algo opcional.
La impresión de la cita/receta no se ha reflejado porque es una acción JavaScript que se realiza contra el navegador.



- Figura 67: Diagrama de secuencia generar diagnóstico y seguimiento de Sintrom -

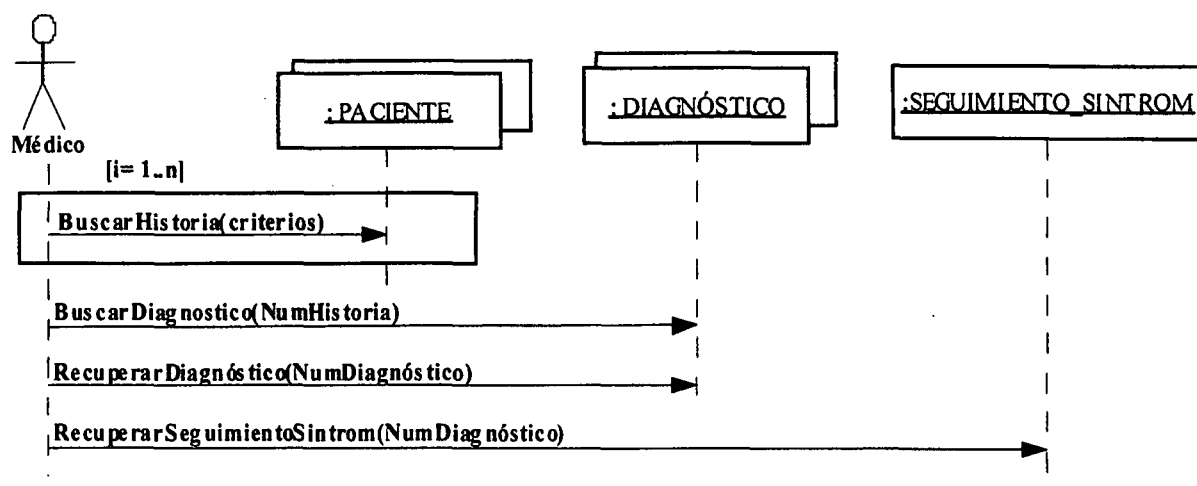


- **Consultar medicamento y modificar medicamento:** Modificar medicamento es algo opcional que puede o no realizar el administrador, si lo desea puede simplemente consultar los datos de un medicamento y terminar así sin realizar ninguna otra acción.



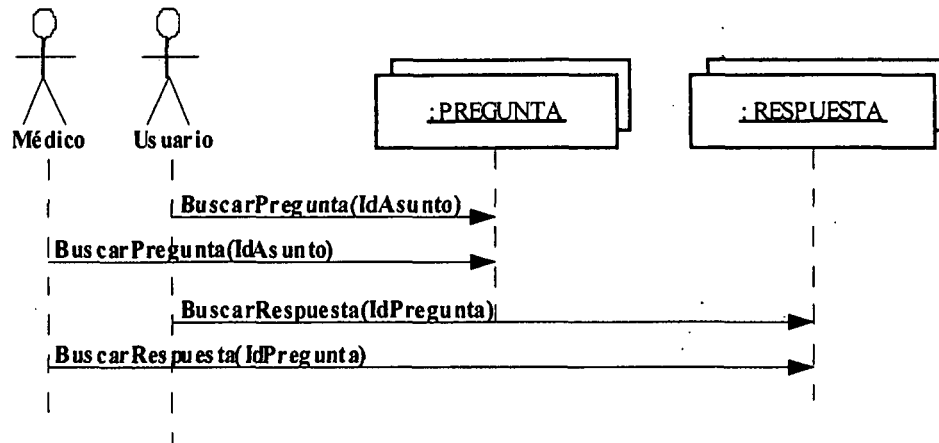
- Figura 68: Diagrama de secuencia consultar y modificar medicamento -

- **Consultar historial:** Si existe un seguimiento de Sintrom en el diagnóstico de un paciente, opcionalmente se podrá consultar.



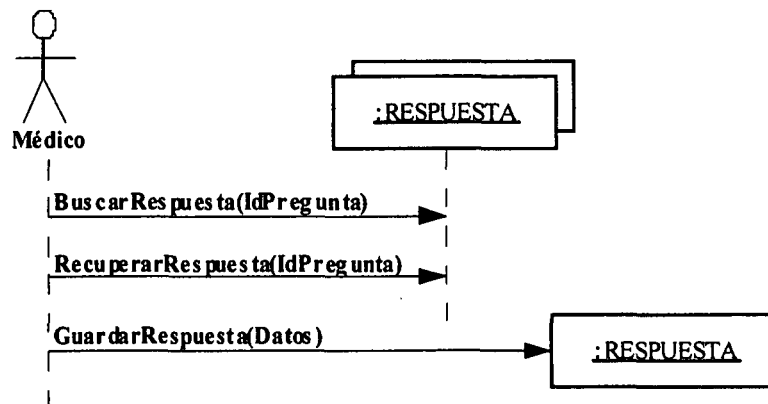
- Figura 69: Diagrama de secuencia consultar historial -

- **Consultar foro:** Únicamente el médico ó usuario deberán seleccionar el asunto del foro que desean consultar apareciendo así todas las preguntas y respuestas asociadas a dicho asunto.



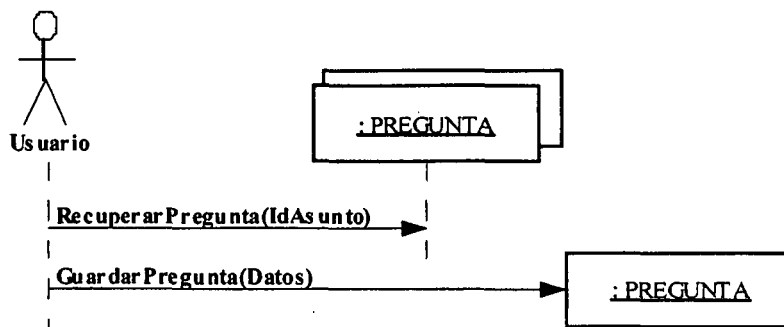
- Figura 70: Diagrama de secuencia consultar foro -

- *Responder al foro*



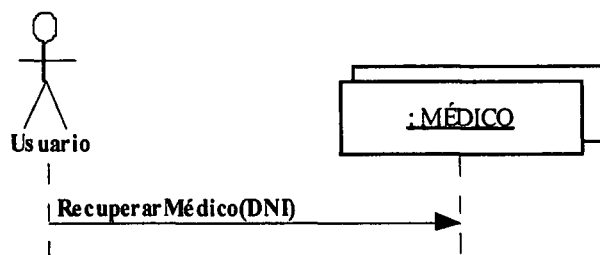
- Figura 71: Diagrama de secuencia responder al foro -

- *Preguntar en el foro*



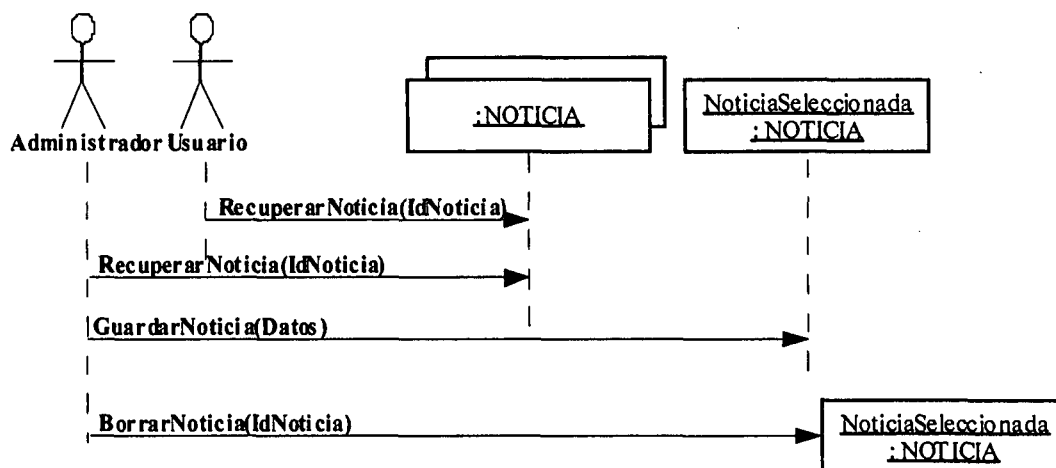
- Figura 72: Diagrama de secuencia preguntar en el foro -

- **Visualizar directorio**



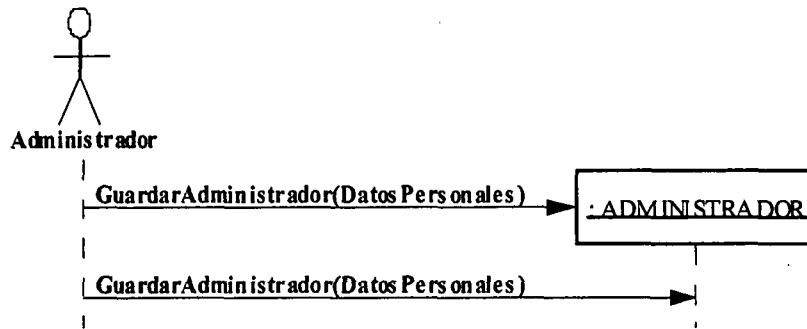
- Figura 73: Diagrama de secuencia visualizar directorio -

- **Consultar noticia:** Opcionalmente el administrador puede modificar la noticia consultada y/o borrarla.



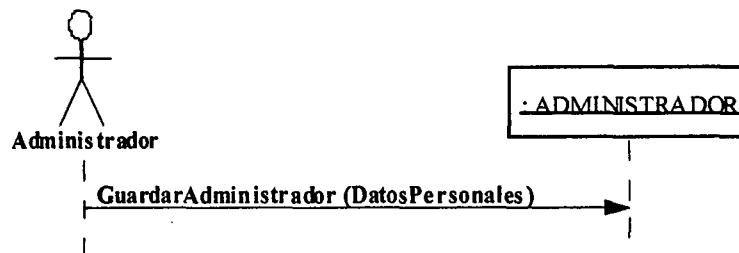
- Figura 74: Diagrama de secuencia consultar noticia -

- **Consultar calendario de vacunaciones:** Simplemente tras pulsar un botón se mostrará el calendario de vacunaciones.
- **Visualizar mapa del centro:** Simplemente tras pulsar un botón se mostrará el mapa del centro.
- **Dar de alta administrador y modificar administrador:** Modificar administrador es opcional.



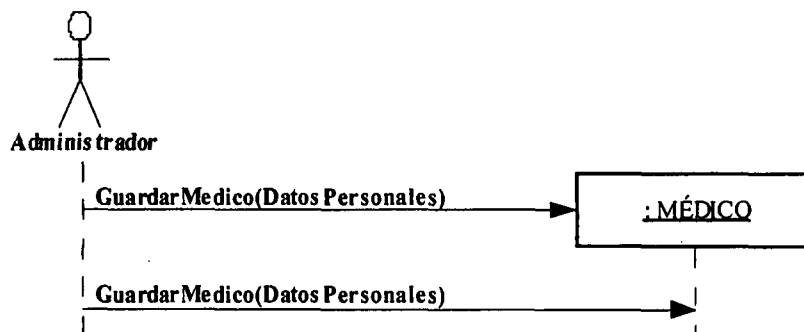
- Figura 75: Diagrama de secuencia dar de alta administrador -

- **Modificar perfil**



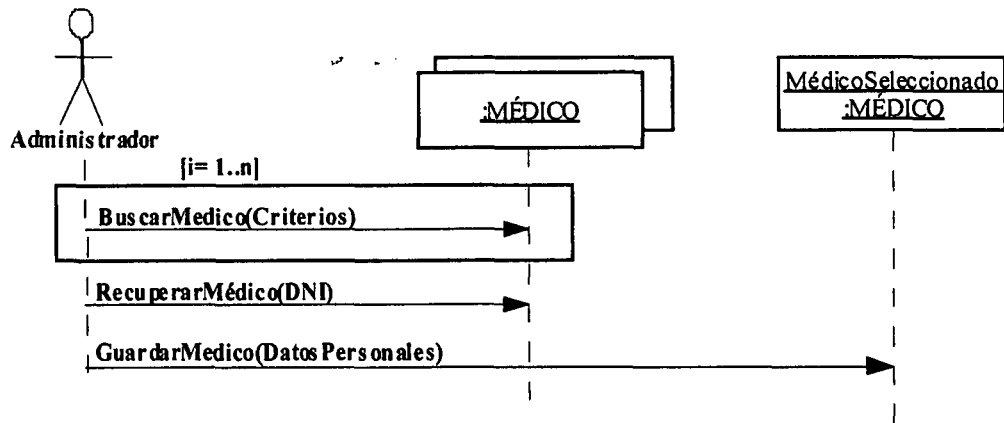
- Figura 76: Diagrama de secuencia modificar perfil -

- **Dar de alta médico y modificar datos médico:** Modificar los datos del médico, es opcional.



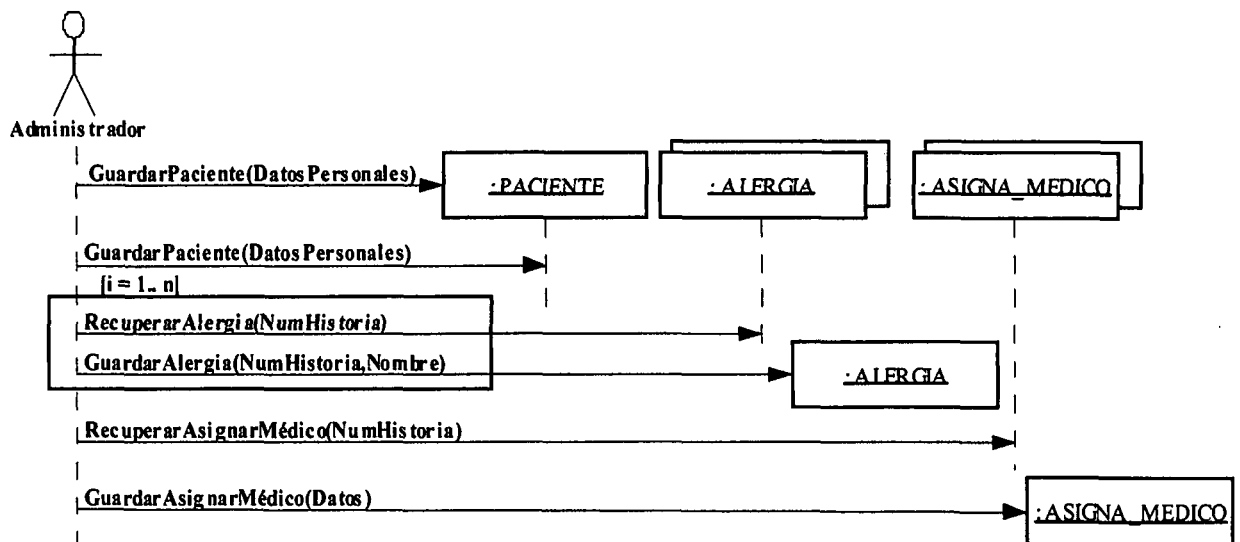
- Figura 77: Diagrama de secuencia dar alta médico -

- **Consultar datos personales médico y modificar médico:** Modificar los datos de un médico es algo opcional que puede o no realizar el administrador, si lo desea puede simplemente consultar los datos personales de un médico y terminar así sin realizar ninguna otra acción.



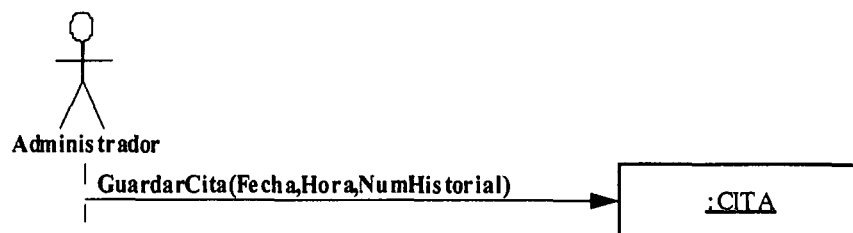
- Figura 78: Diagrama de secuencia consultar datos personales y modificar médico -

- **Dar de alta paciente y modificar paciente:** Modificar el paciente es opcional, al igual que añadir/modificar alergias y asignar un médico.



- Figura 79: Diagrama de secuencia dar de alta paciente -

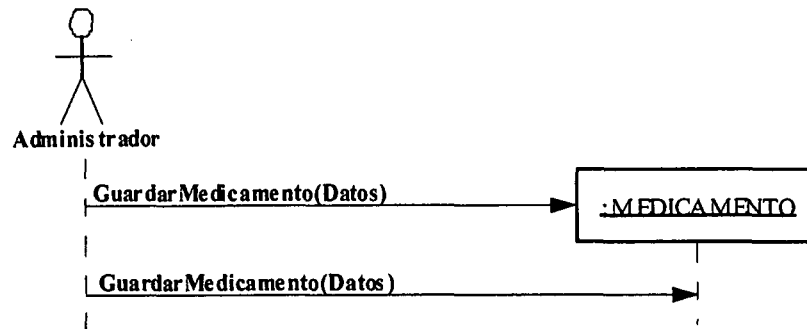
- **Generar cita**



- Figura 80: Diagrama de secuencia generar cita -

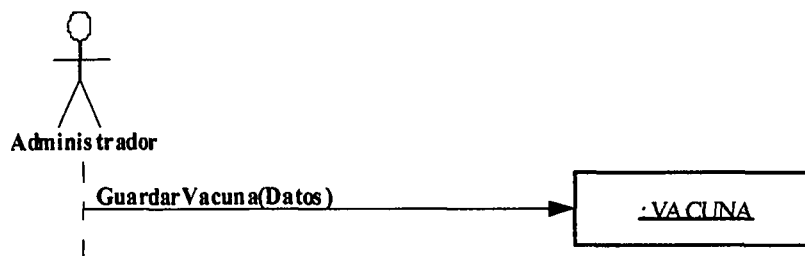


- **Dar de alta medicamento y modificar medicamento:** Modificar medicamento es opcional.



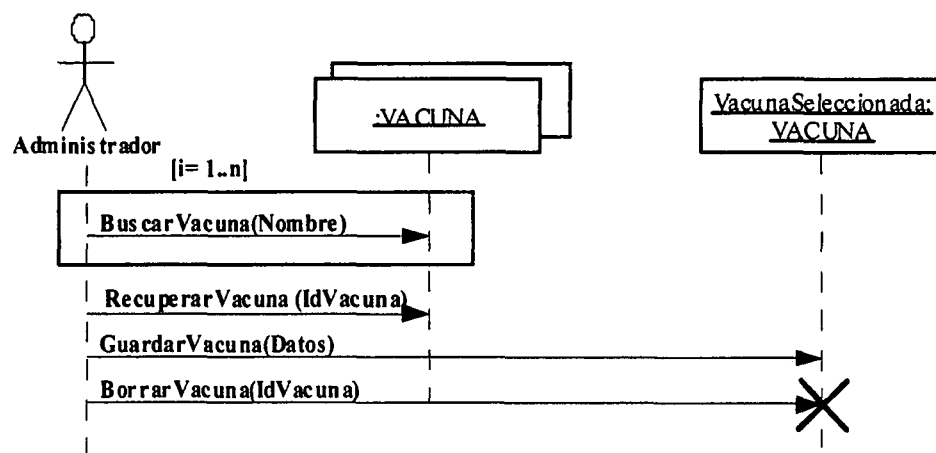
- Figura 81: Diagrama de secuencia dar de alta medicamento -

- **Dar alta vacuna**



- Figura 82: Diagrama de dar alta vacuna -

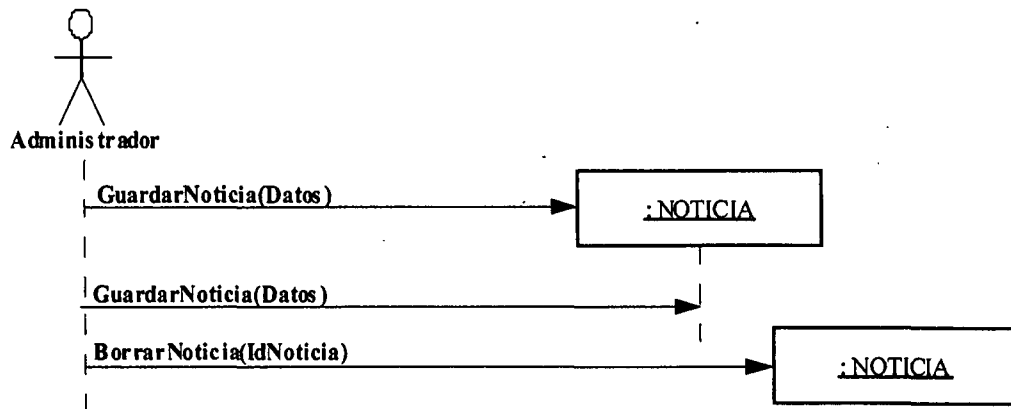
- **Cambiar calendario de vacunaciones:** En este caso tanto modificar vacuna, como borrarla son opcionales, así que se puede realizar solo una de las acciones, ninguna, o las dos, teniendo en cuenta que modificar se tiene que realizar antes de borrar.



- Figura 83: Diagrama de secuencia cambiar calendario de vacunaciones -

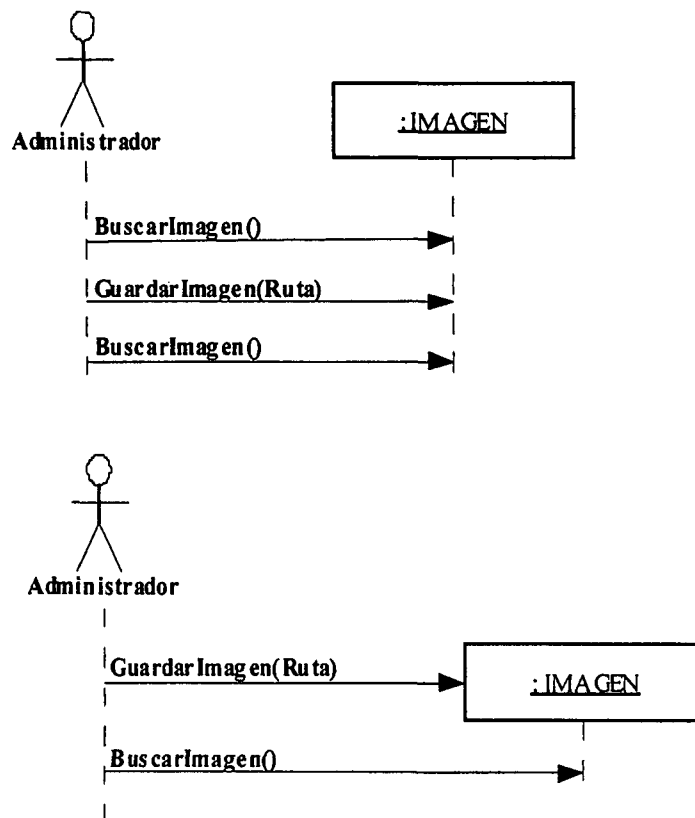


- **Dar alta noticia y modificar/borrar noticia:** Modificar/borrar noticia es opcional.



- Figura 84: Diagrama de secuencia dar alta noticia -

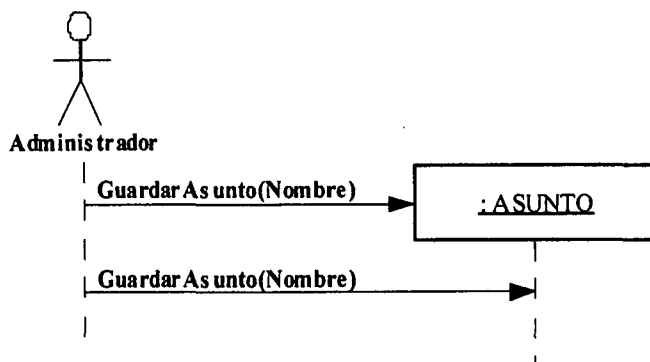
- **Cambiar mapa del centro:** Se muestran dos diagramas de secuencia para cambiar el mapa del centro. El primero de ellos, refleja el cambio cuando ya existe una imagen, por lo tanto antes del cambio y después se permite opcionalmente visualizar la imagen. El segundo, se da cuando no existe una imagen previa.



- Figura 85: Diagramas de secuencia cambiar mapa del centro -

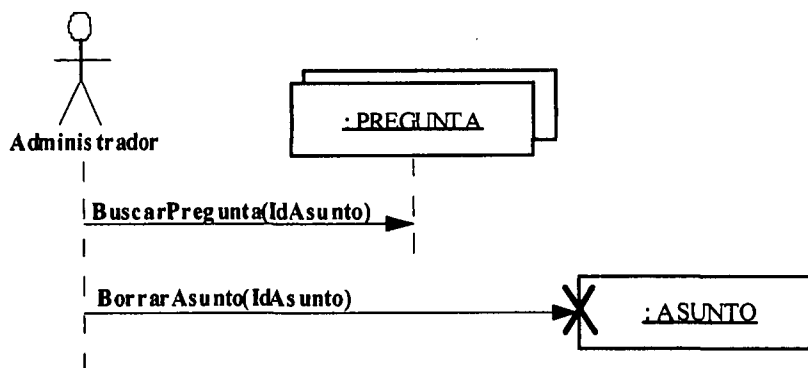


- **Crear asunto y modificar asunto:** Modificar asunto es opcional.



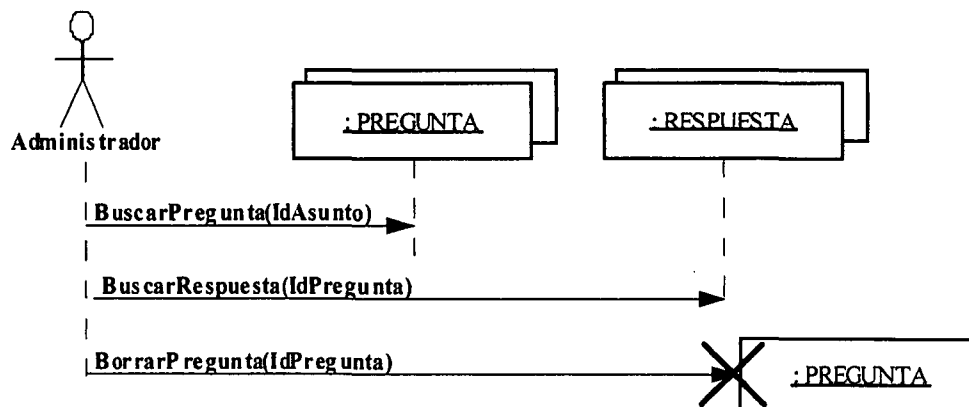
- Figura 86: Diagrama de secuencia crear asunto -

- **Borrar asunto:** Como se puede observar, antes de borrar se deben consultar las preguntas que existen en el asunto.



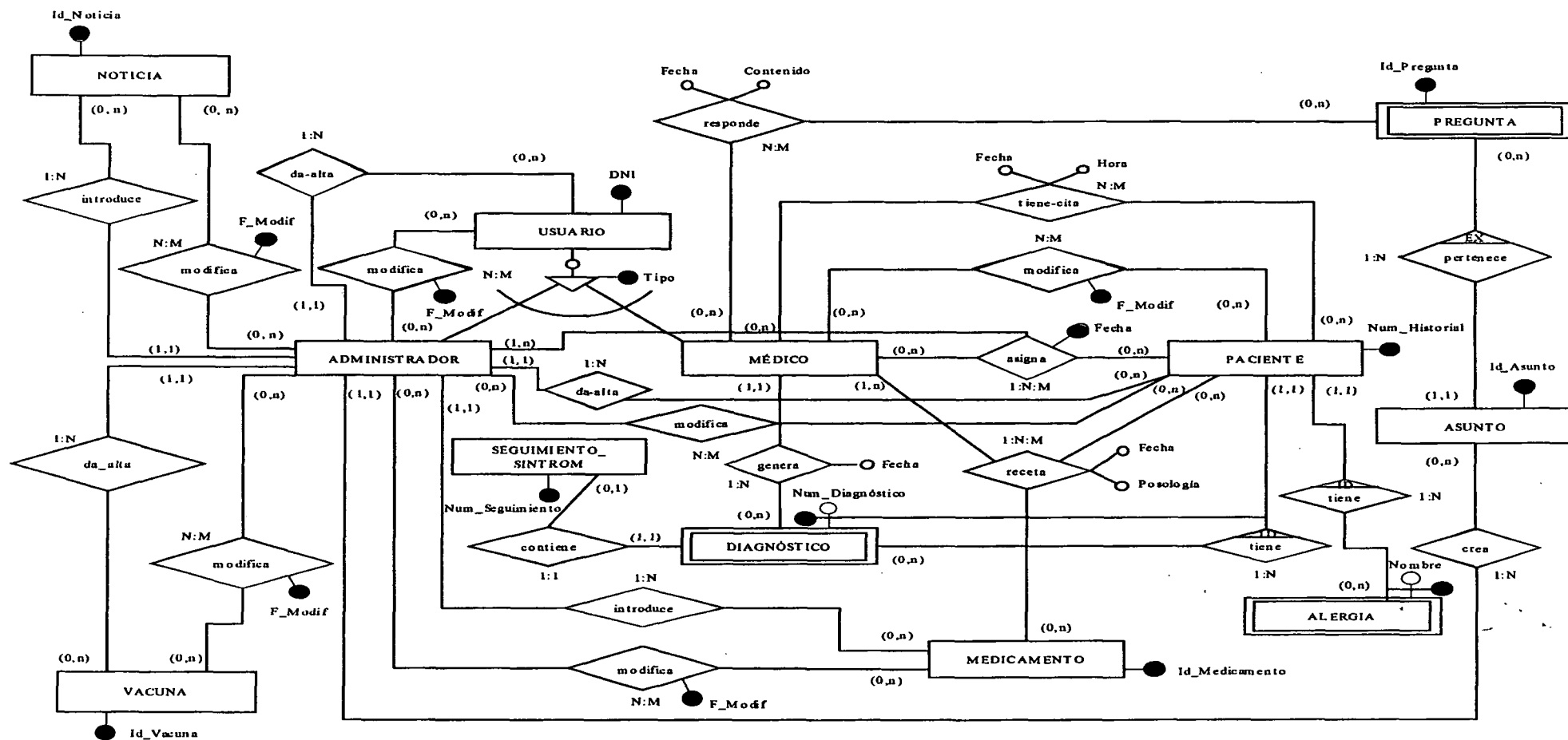
- Figura 87: Diagrama de secuencia borrar asunto -

- **Borrar pregunta:** Como se puede observar, antes de borrar se deben consultar las respuesta que existen en esa pregunta.



- Figura 88: Diagrama de secuencia borrar pregunta -

4.6.- DIAGRAMA E/R



- Figura 89: Diagrama E/R -



El diagrama entidad-relación ó diagrama E/R, es una herramienta para el modelado de datos de un sistema de información. Mediante este diagrama expresamos las entidades que serán relevantes en nuestra aplicación, sus interrelaciones y por supuesto, sus propiedades, a través de un lenguaje gráfico.

Los principales elementos de cualquier diagrama E/R son:

- Las entidades, que son personas, lugares, cosas, conceptos o sucesos reales o abstractos que contienen información de interés para la aplicación. Se representan mediante un rectángulo etiquetado en su interior mediante un nombre.
- Los atributos, que son las propiedades relevantes de una entidad o una interrelación.
- Las interrelaciones, que son las asociaciones entre entidades. Se representan mediante un rombo etiquetado en su interior mediante un verbo

Por motivos de legibilidad, no se han representado todos los atributos de las entidades a excepción de sus claves primarias, los demás atributos están reflejados en el modelo relacional.

4.6.1.- SUPUESTOS SEMÁNTICOS

- Un administrador introduce o no una noticia en la Web, y una noticia solo puede ser introducida por un administrador.
- Un administrador puede o no modificar noticias, y una noticia puede ser modificada por ninguno o varios administradores.
- Un administrador puede o no dar de alta vacunas, y una vacuna es dada de alta por un administrador.
- Un administrador puede o no modificar vacunas, y una vacuna puede ser modificada por cero o varios administradores.
- Un administrador puede o no dar de alta usuarios y un usuario es dado de alta por un administrador. El primer administrador de la BD, no es dado de alta por ningún administrador sino que es el propio sistema quien lo da de alta.
- Un administrador puede o no modificar usuarios, y un usuario puede ser o no modificado por distintos administradores.
- Se debe controlar que un administrador solo es modificado por si mismo tantas veces como desee, pero nunca puede modificarle otro administrador, a excepción que el administrador que le ha dado de alta, y justo en ese momento para corregir algún error.
- Un administrador puede o no introducir medicamentos, y un medicamento es introducido por un único administrador.
- Un administrador puede o no modificar medicamentos, y un medicamento puede ser modificado o no por distintos administradores.
- Un administrador puede o no crear asuntos en el foro, y un asunto es creado por un único administrador.
- Un administrador puede o no dar de alta pacientes, y un paciente es dado de alta por un único administrador.



- Un administrador puede modificar o no pacientes, y un paciente es modificado por ninguno o varios administradores.
- Un administrador asigna a un paciente cero o varios médicos, un administrador asigna un médico a cero o varios pacientes, y un médico es asignado a un paciente por uno o varios administradores.
- Un médico puede o no modificar pacientes, y un paciente es modificado por cero o varios médicos.
- Un médico puede o no citar pacientes, y un paciente puede o no ser citado por distintos médico. Solo lo puede citar el médico que lo tiene asignado en ese momento.
- Un médico le receta a un paciente cero o varios medicamentos, un medico receta un medicamento a cero o varios pacientes, y a un paciente le receta un medicamento uno o varios médicos.
- Un diagnóstico necesita para poder identificarse el número de historial al que pertenece, ya que el historial del paciente estará lleno de diagnósticos.
- La entidad alergia es débil en identidad respecto al paciente, ya que para poder identificarse necesita el número de historia del paciente. Esto es necesario para que en caso de desaparecer un paciente, desaparezcan también sus alergias.
- Un paciente puede o no tener alergias, y una alergia pertenece a un único paciente.
- Un paciente puede o no tener diagnósticos, y un diagnóstico solo pertenece a un paciente.
- Un médico puede o no generar diagnósticos, y un diagnóstico es generado por un único médico.
- Un diagnóstico puede contener uno ó ningún seguimiento de Sintrom, y un seguimiento de Sintrom pertenece a un único diagnóstico.
- Una pregunta es débil en existencia respecto a asunto, ya que si no hay asunto no puede haber pregunta.
- Una pregunta pertenece a un asunto, y en un asunto puede o no haber preguntas.
- Un médico puede o no responder a preguntas, y una pregunta puede o no ser respondida por los médicos.
- Un usuario puede ser o un médico o un administrador, pero no ambas cosas a la vez.

4.7.- MODELO RELACIONAL

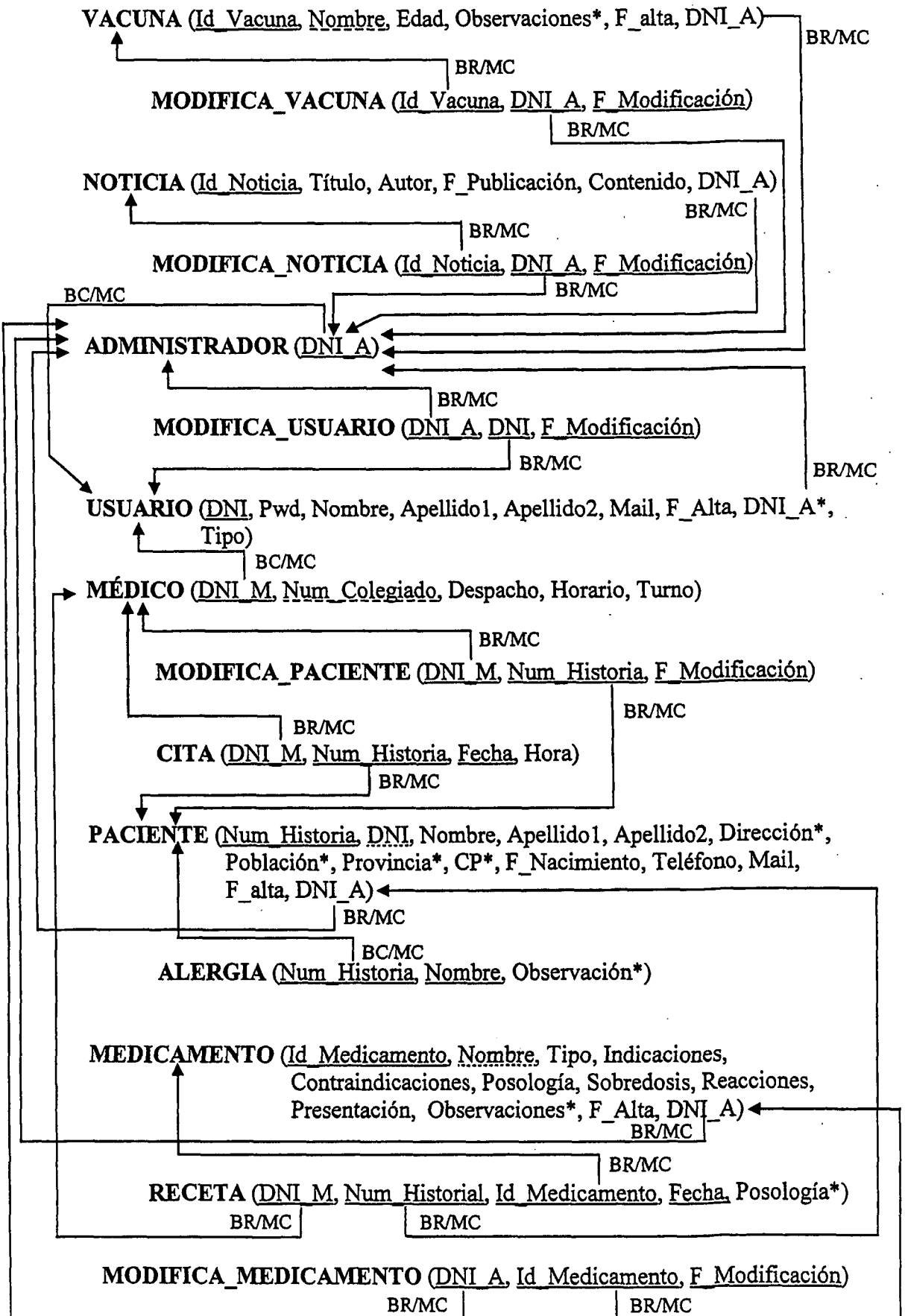
El modelo relacional organiza y representa los datos reflejados en el diagrama E/R, en forma de tablas o relaciones.

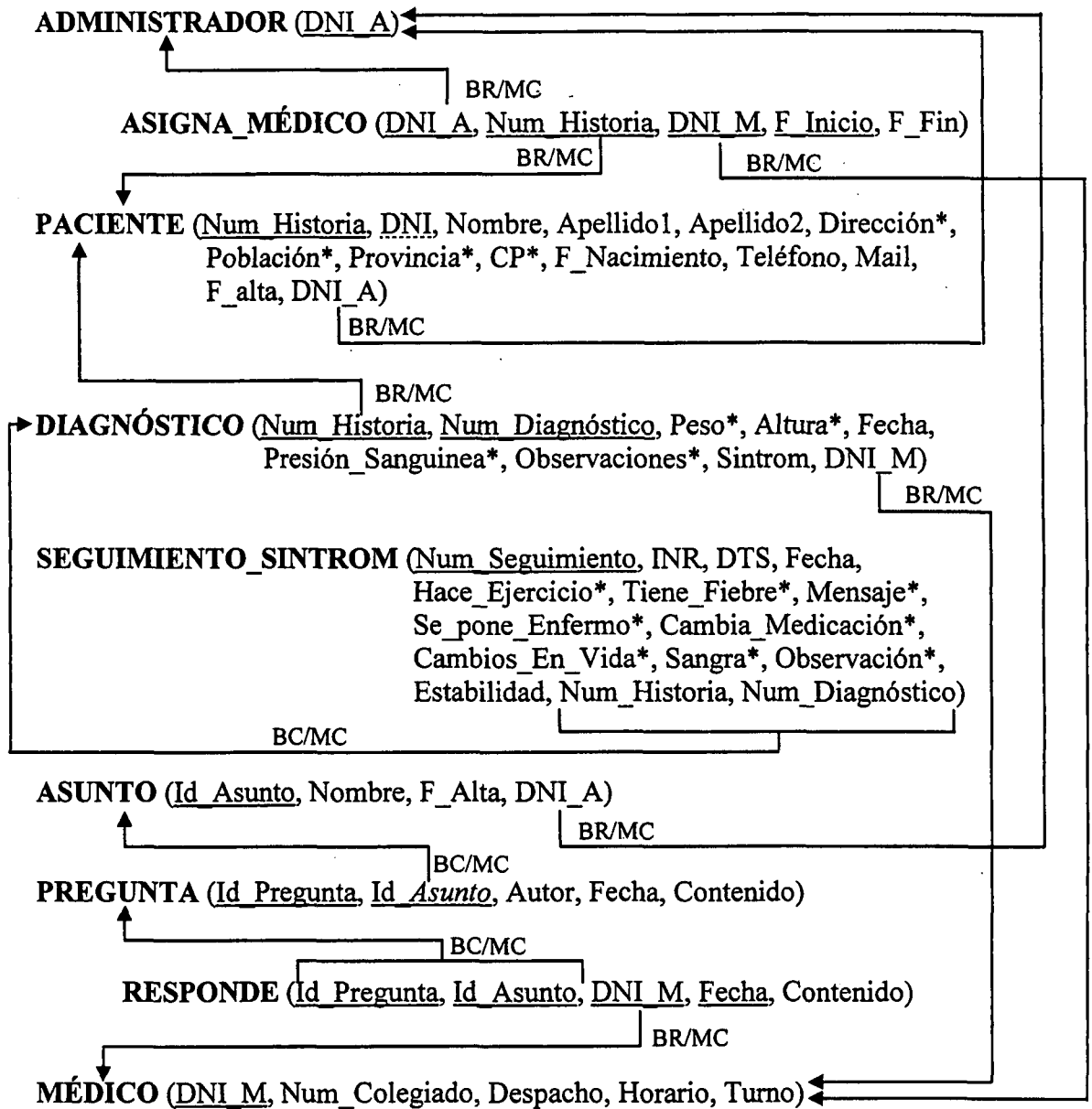
Una relación representa una tabla, donde cada fila representa una colección de valores que describen una entidad del mundo real. Cada fila se denomina tupla o registro y cada columna campo.

En este modelo, se han subrayado las claves primarias, se han subrayado en discontinuo los atributos alternativos y se ha puesto un asterisco en aquellos atributos que pueden tomar valor nulo.



En cuanto a los borrados y modificaciones, como en la aplicación no existe ninguna opción en la que se pueda borrar nada a excepción de *borrar vacuna*, *asunto* y *pregunta*, se considerará que todos los borrados y modificaciones son restringidos, es decir, que si existen tuplas en las relaciones hijas relacionadas con la tupla de la relación padre no se permitirá realizar la operación. Las opciones de *borrar vacuna*, *asunto* y *pregunta*, no están contempladas en este modelo, ya que no necesitamos almacenar ningún histórico sobre ellas.







5.- PROTOTIPO: INSTALACIÓN Y MANUAL DE USUARIO

5.1. INSTALACIÓN

En este apartado se describirá con alto nivel detalle, los pasos a seguir para realizar la instalación de cada uno de los programas y utilidades necesarias para el correcto funcionamiento de la aplicación en un ordenador.

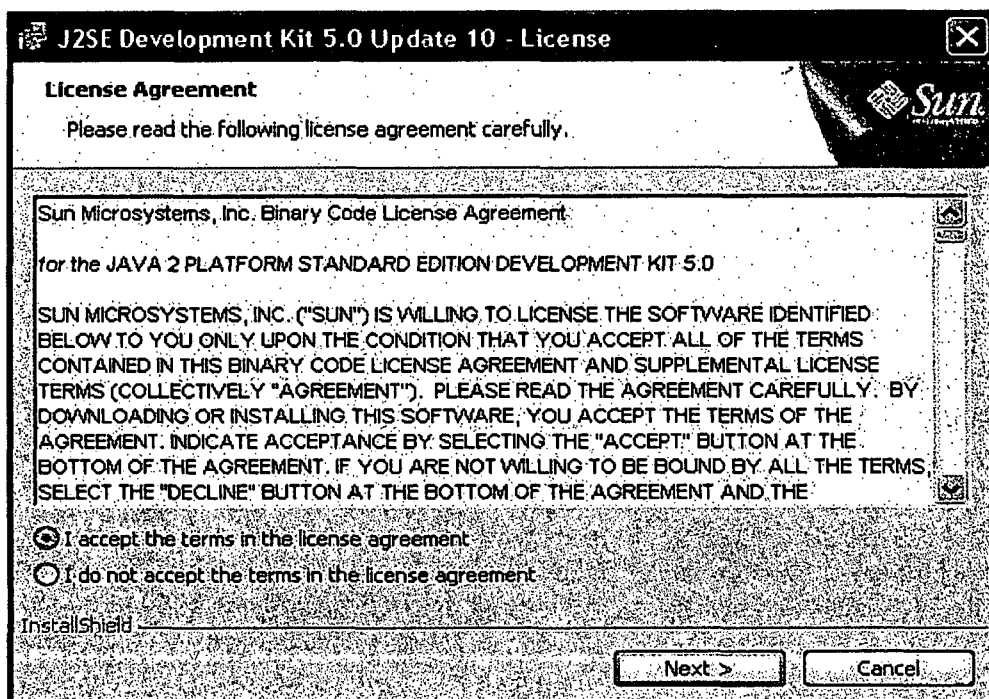
5.1.1.- INSTALACIÓN DE JDK

Para descargar el archivo ejecutable de JDK 5.0 Update 10, introducir la siguiente dirección http://java.sun.com/javase/downloads/index_jdk5.jsp en un navegador, y descargar el archivo “*Java Developer Kit 5.0 Update 10*”.

Este archivo contiene la instalación completa de la Máquina Virtual de Java, conocida como, Java Runtime Environment (JRE), mediante la cual se proporcionan las librerías necesarias para compilar programas escritos en Java. Por otro lado, contiene el Java Developer Kit (JDK), el cual incorpora un compilador y un conjunto de herramientas de desarrollo para la creación de programas independientes y applets Java.

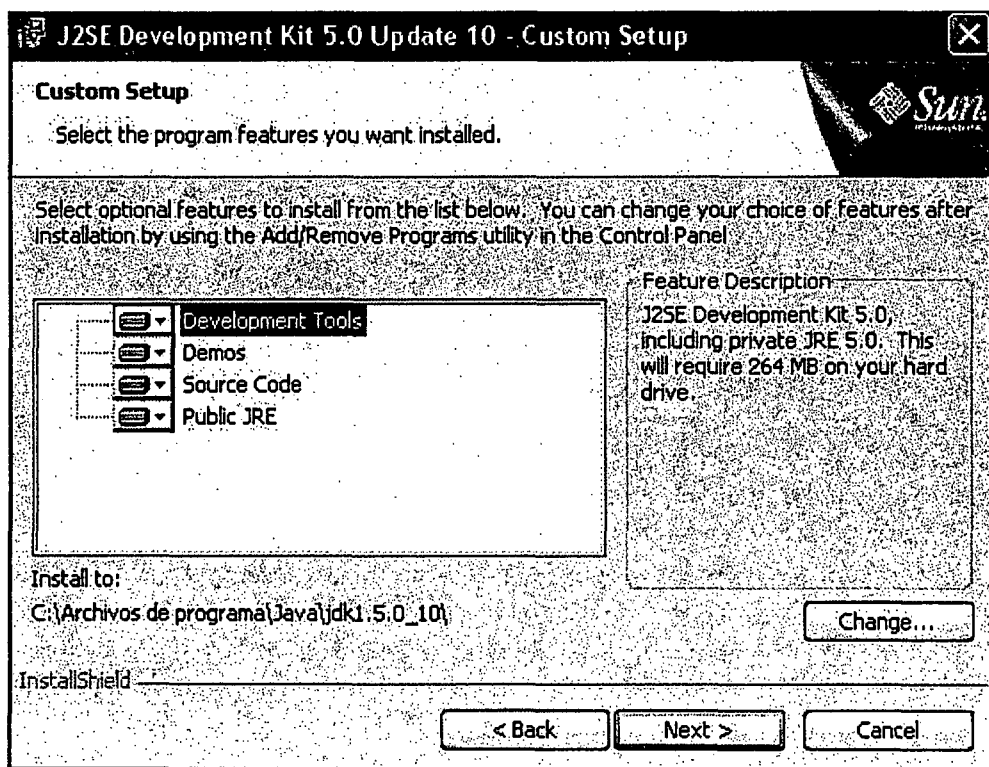
Una vez descargado el archivo lo ejecutaremos y seguiremos los siguientes pasos:

1. Leemos y aceptamos el contrato de licencia del programa, y pulsamos “Next”.



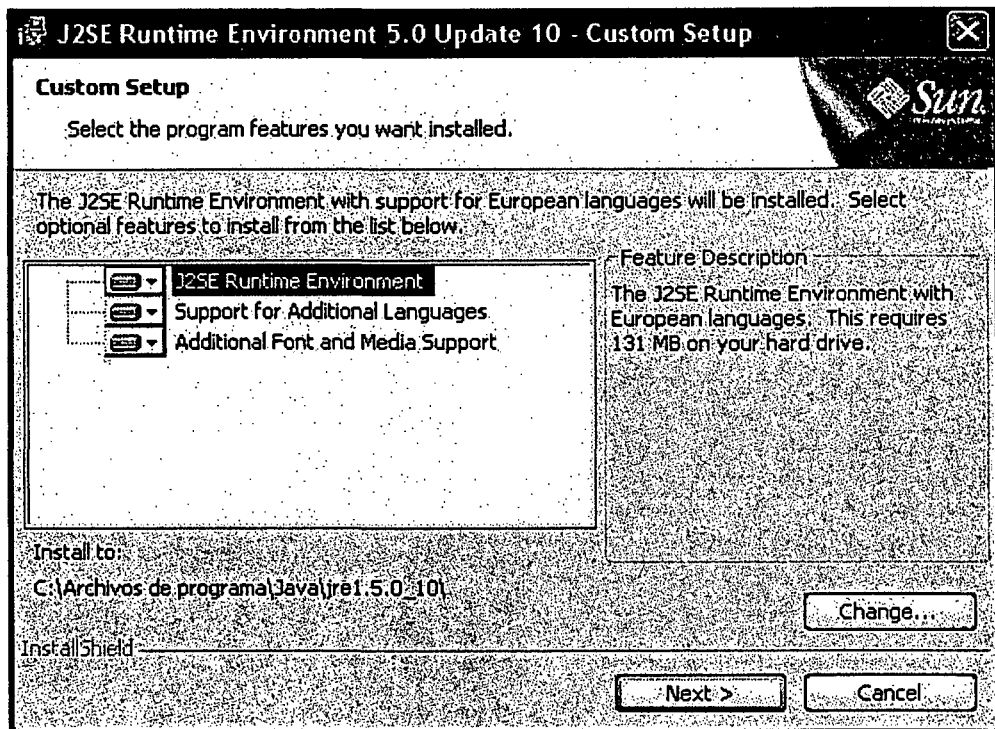
- Figura 90: Instalación JDK 1 -

2. Ahora se instalará el compilador (JDK). Dejamos la instalación por defecto, por lo tanto solo debemos pulsar "Next".



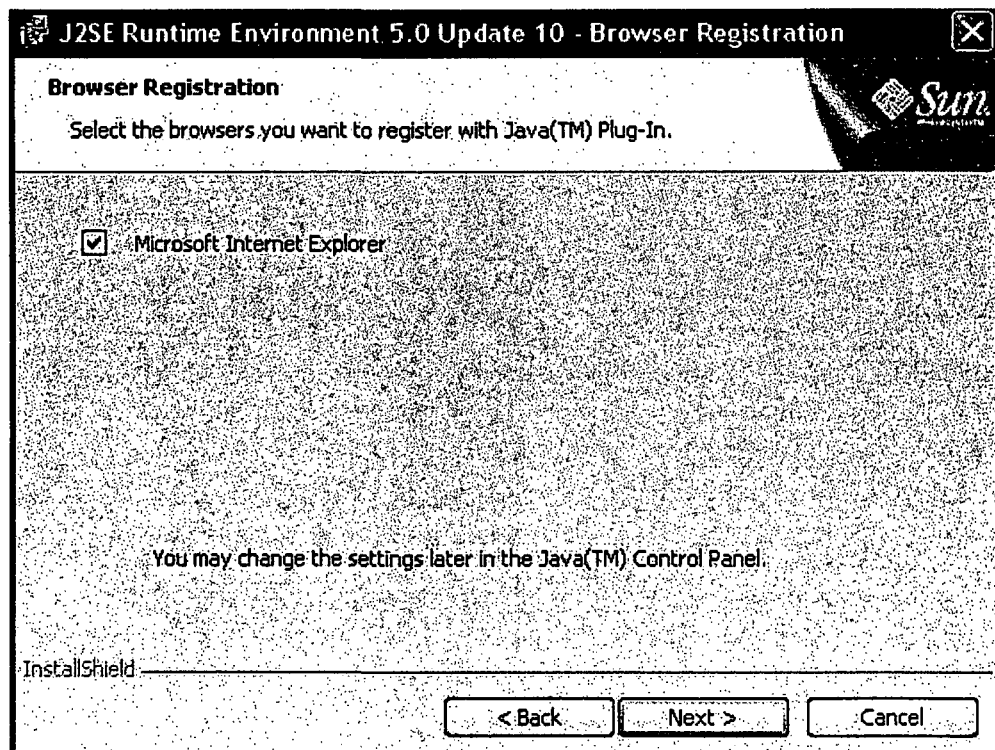
- Figura 91: Instalación JDK 2 -

3. Ahora se instalará la Máquina Virtual de Java (JRE). Volvemos a dejar la instalación por defecto pulsando "Next".



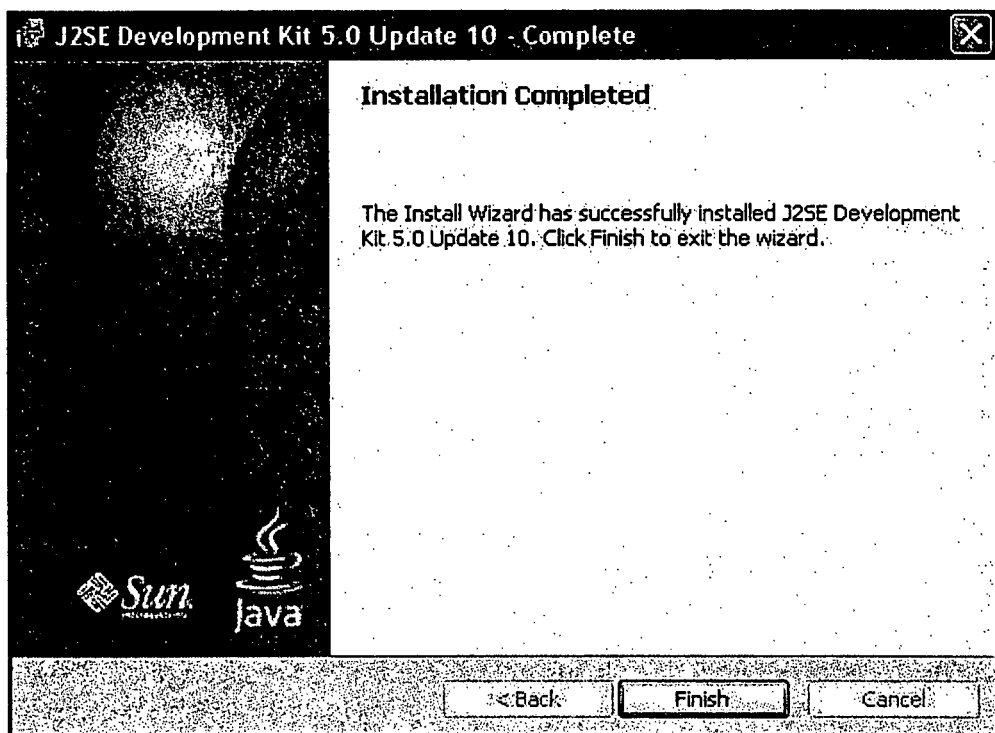
- Figura 92: Instalación JDK 3 -

4. Registramos el navegador que deseamos que funcione con Java, en este caso seleccionamos "Microsoft Internet Explorer" ignorando cualquier otro navegador que aparezca en la selección. Después pulsamos "Next".



- Figura 93: Instalación JDK 4 -

5. Una vez se haya completado la instalación, pulsaremos "Finish".





- Figura 94: Instalación JDK 5 -

5.1.2.- INSTALACIÓN DE MYSQL 5.0

Para descarga el archivo ejecutable, introduciremos la siguiente dirección <http://dev.mysql.com/downloads/> en el navegador. Una vez en la página pulsaremos sobre el botón “Download” de MySQL Community Server (*Ver imagen*).

MySQL Downloads

I Can Do It Myself	I Want Enterprise-Grade Software, Services & Support
I am comfortable upgrading and configuring MySQL	MySQL Enterprise Server and the Configuration Wizard make upgrading and deploying MySQL easy
I can resolve technical issues myself	MySQL Enterprise Support provides fast answers to your technical questions
I have DBA skills	MySQL Certified DBAs Available 24x7 to provide consultative advice
I am familiar with database security best practices	MySQL Security Advisor continuously monitors your system for security vulnerabilities
I understand how to design and set-up MySQL replication	MySQL Replication Advisor enables you to improve replication design and performance
I know how to improve MySQL performance	Optimize performance using the MySQL Performance Advisor
<i>I am fluent in open source - take me straight to the packages!</i>	<i>Tell me more about how MySQL can help me develop, deploy and manage my database applications</i>
MySQL Community Server  Download »	MySQL Enterprise  Learn More »

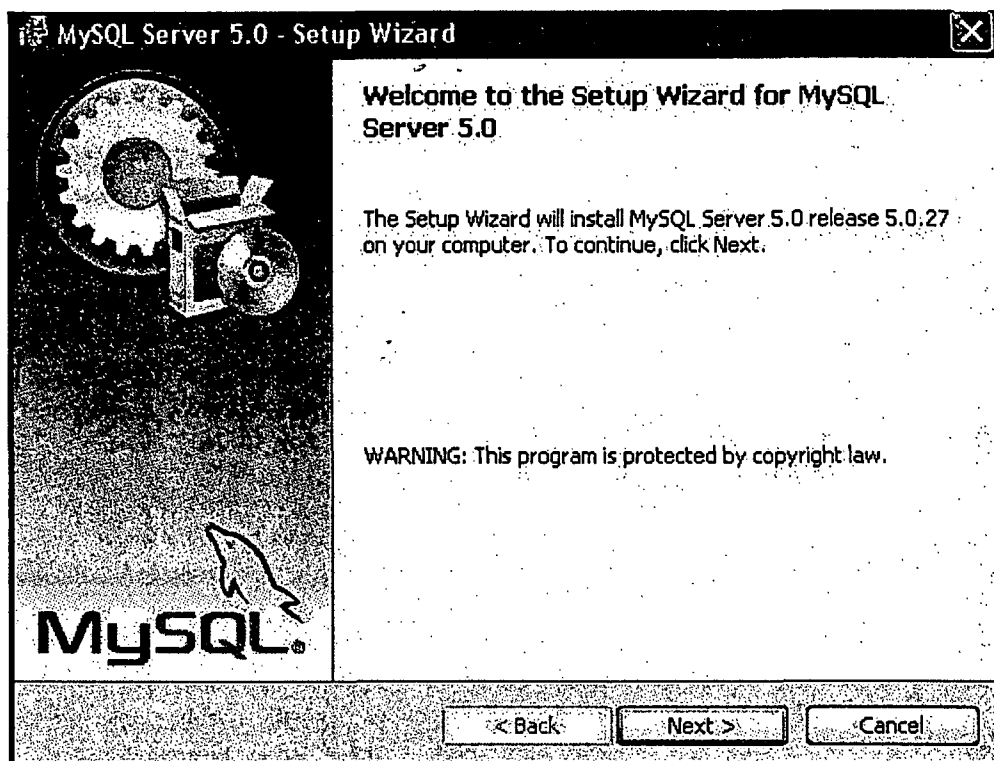
Pulsar aquí

- Figura 95: Descargar MySQL -

Después seleccionaremos el instalable de la plataforma Windows “*Windows (x86) ZIP/Setup.EXE*”.

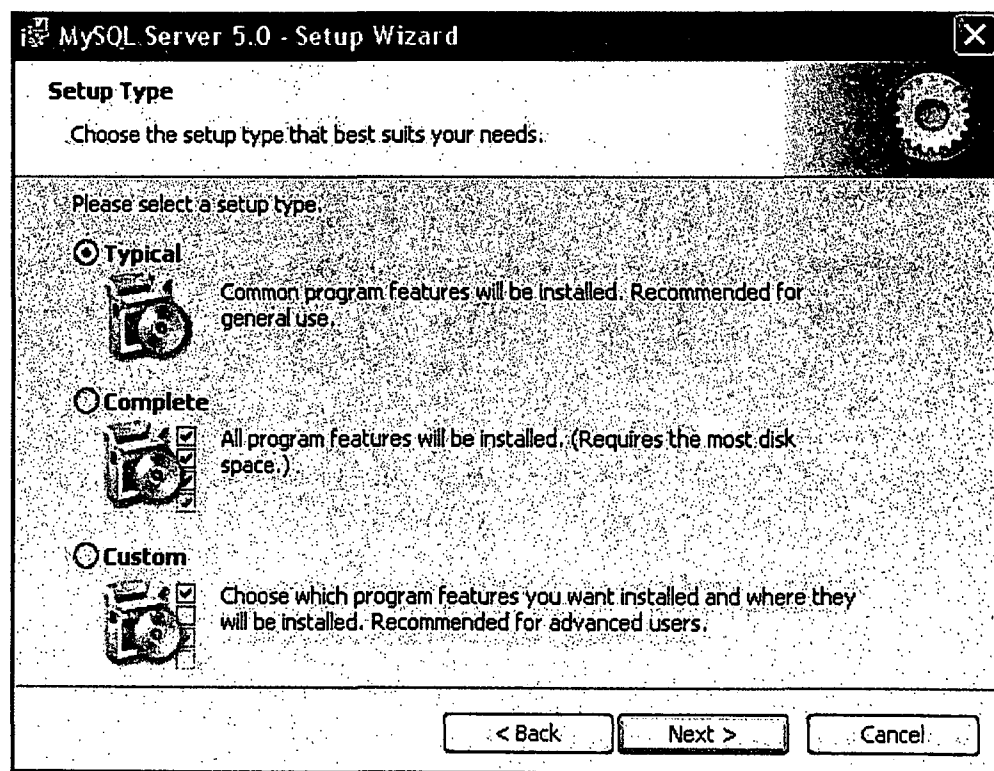
Una vez descargado el paquete deberemos descomprimirlo. Se creará una carpeta que dentro contiene un único archivo, “*Setup.exe*”. Ejecutaremos ese archivo y comenzaremos la instalación siguiendo los siguientes pasos:

1. Primeramente aparecerá una pantalla de bienvenida que nos advertirá que el programa está protegido con Copyright. Pulsamos “Next”.



- Figura 96: Instalación de MySQL 1 -

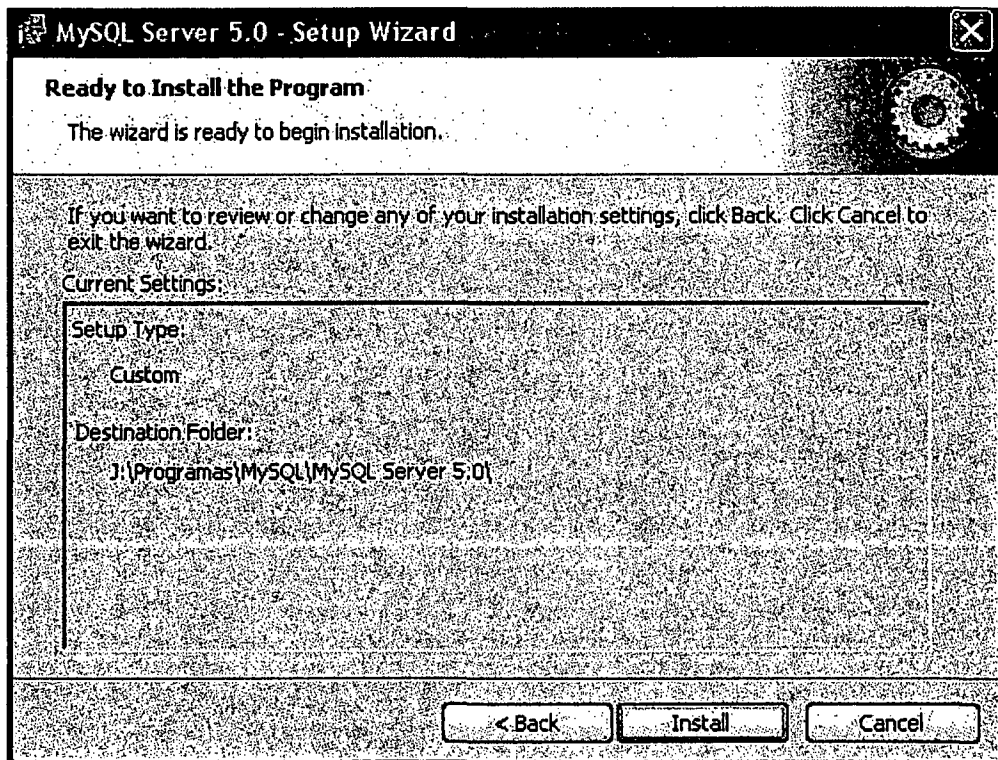
2. La siguiente pantalla nos pedirá que seleccionemos el tipo de instalación que deseamos. En este caso seleccionamos la instalación típica y pulsamos "Next".



- Figura 97: Instalación de MySQL 2 -



3. En la siguiente pantalla nos indicará el tipo de instalación y la ruta donde va a instalarse. Pulsamos “Install” para que comience el asistente de instalación.

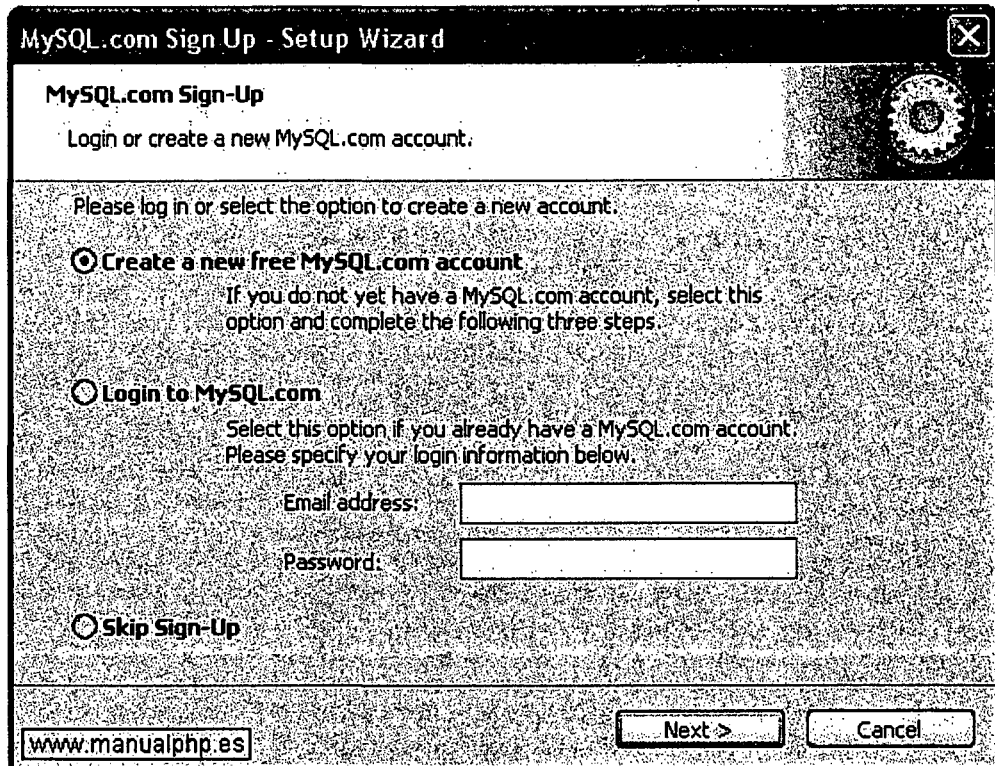


- Figura 98: Instalación de MySQL 3 -

4. Tras unos minutos aparecerá la pantalla de registro. Esta pantalla tiene tres opciones:

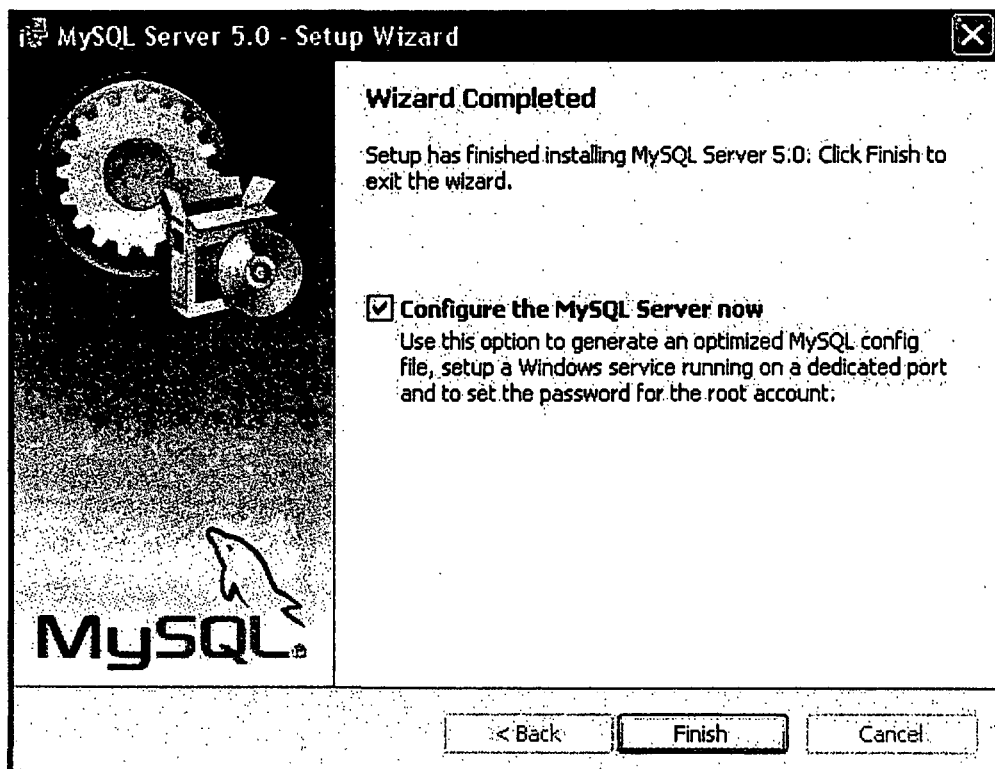
- Crear una nueva cuenta gratis en MySQL.com.
- Entrar con tu cuenta.
- Saltar el registro.

En este caso seleccionaremos saltar el registro y pulsaremos “Next”.



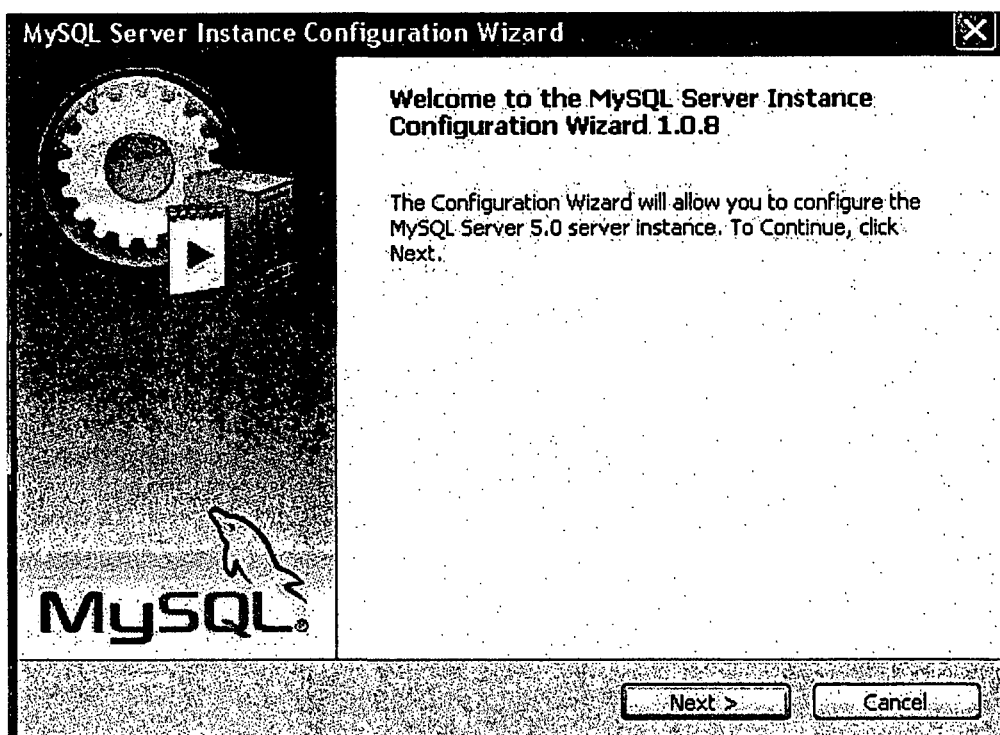
- Figura 99: Instalación de MySQL 4 -

5. Una vez finalizada la instalación, deberemos configurar MySQL. Marcaremos la opción "*Configure the MySQL Server now*", para que se abra el asistente de configuración, después pulsaremos "Finish".



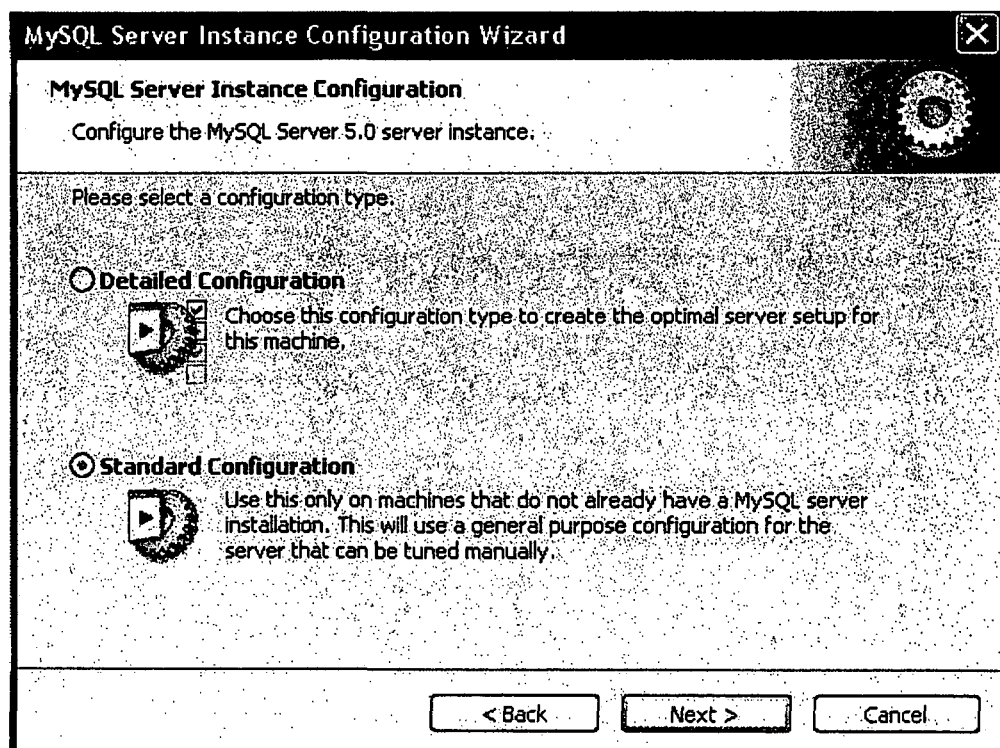
- Figura 100: Instalación de MySQL 5 -

6. Aparecerá una pantalla que te da la bienvenida al asistente de configuración.
 - Pulsamos “Next”.



- Figura 101: Instalación de MySQL 6 -

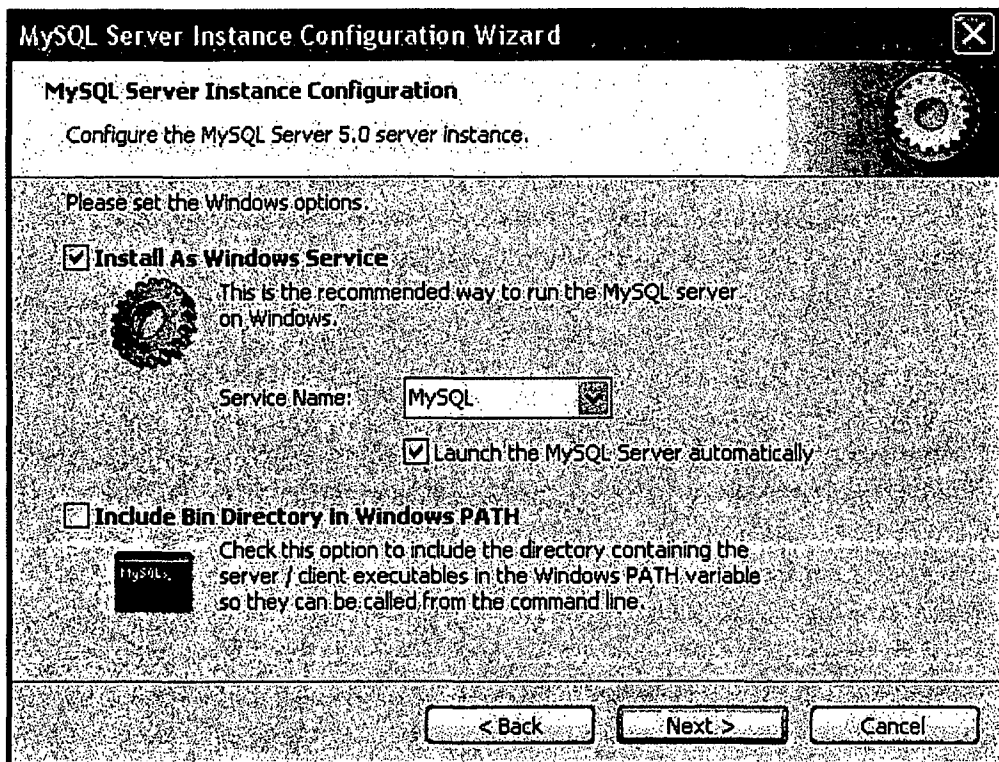
7. Ahora nos preguntará que tipo de configuración deseamos. Seleccionaremos la configuración estándar y pulsaremos “Next”.



- Figura 102: Instalación de MySQL 7 -

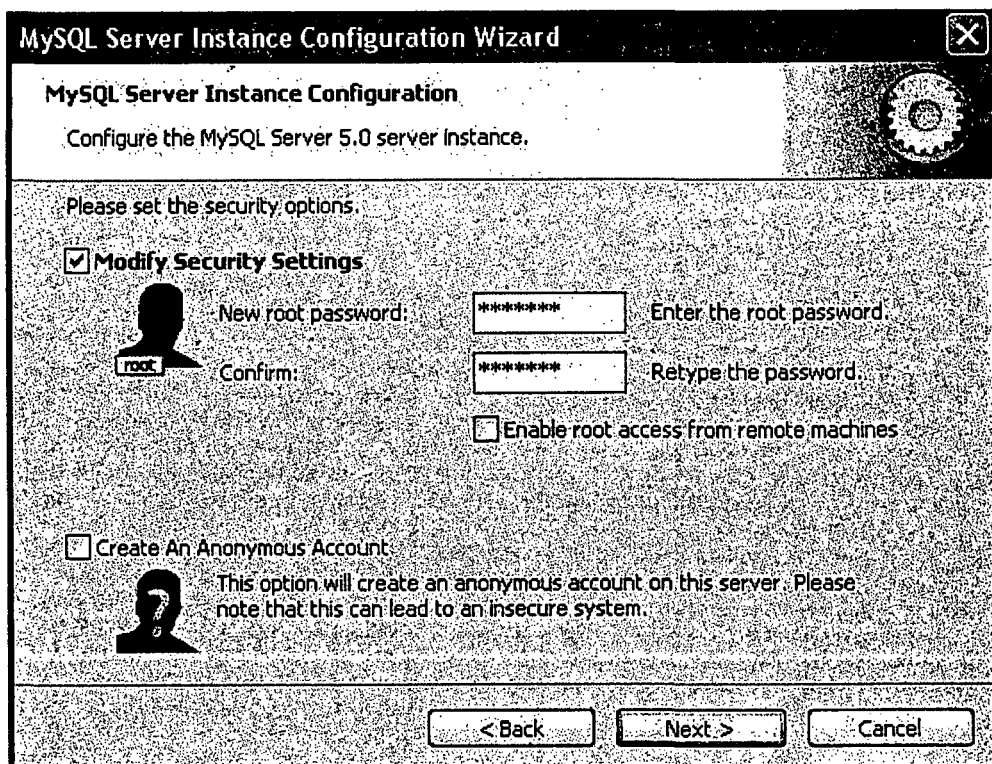


8. La siguiente pantalla configura el servicio de MySQL. Dejamos todos como está y pulsamos "Next".



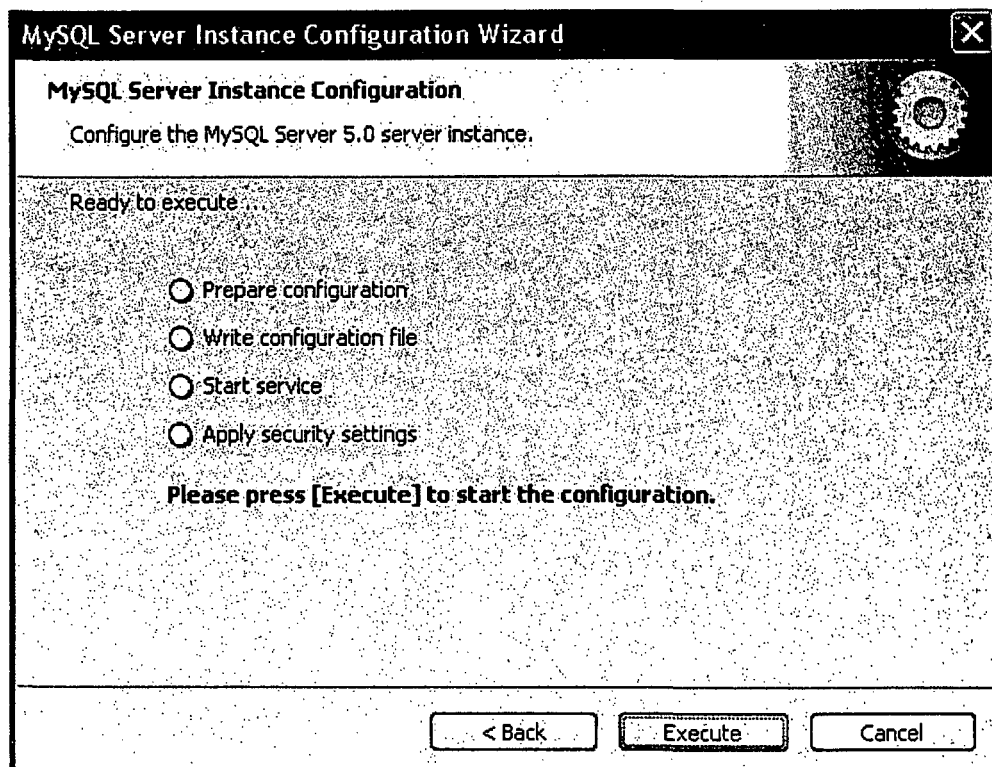
- Figura 103: Instalación de MySQL 8 -

9. Ahora deberemos crear el administrador de la base de datos. Seleccionamos "Modify Security Settings", introducimos la contraseña que deseemos y la confirmamos volviéndola a escribir. Después pulsamos "Next".



- Figura 104: Instalación de MySQL 9 -

10. Una vez hemos terminado de configurar MySQL, aparecerá una pantalla donde se nos pedirá que ejecutemos la configuración para ver si todo está correcto. Pulsamos "Execute".

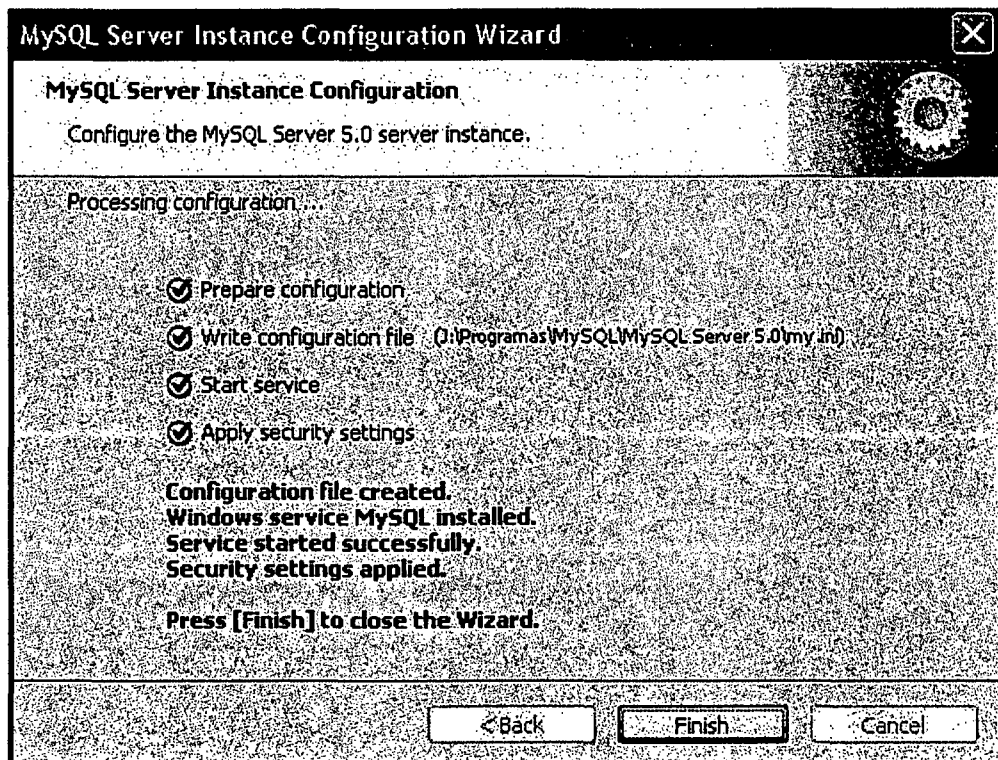


- Figura 105: Instalación de MySQL 10 -



11. Una vez se haya ejecutado todo. Si no ha ocurrido ningún problema, aparecerá la siguiente pantalla, pulsaremos "Finish" y todo el proceso habrá terminado.

En caso de producirse algún error, puede ser debido al firewall, o algún antivirus que bloquee el servicio.



- Figura 106: Instalación de MySQL 11 -

5.1.3.- DESCARGA DEL DRIVER PARA MYSQL 5.0

El driver de la base de datos es un elemento muy importante, ya que gracias a él podremos conectar la base de datos a través de JDBC desde una aplicación Java.

El driver se encargará de convertir la información que se envía a través de la aplicación a un formato que lo entienda la base de datos.

Para descargar el driver de MySQL 5.0, accederemos a esta página <http://dev.mysql.com/downloads/connector/odbc/5.0.html> y seleccionaremos la opción para Windows del driver 5.0 "Windows (x86) ZIP/Setup.EXE".

Una vez descargado descomprimiremos el directorio y de momento no nos preocuparemos, ya que se explicará más adelante que hacer con él.



5.1.4.- INSTALACIÓN DE SQLYOG 5.21

Para descargar el archivo ejecutable, introduciremos la siguiente dirección <http://www.webyog.com> en el navegador. Una vez en la página nos meteremos en "Download" y descargaremos la versión 5.21.

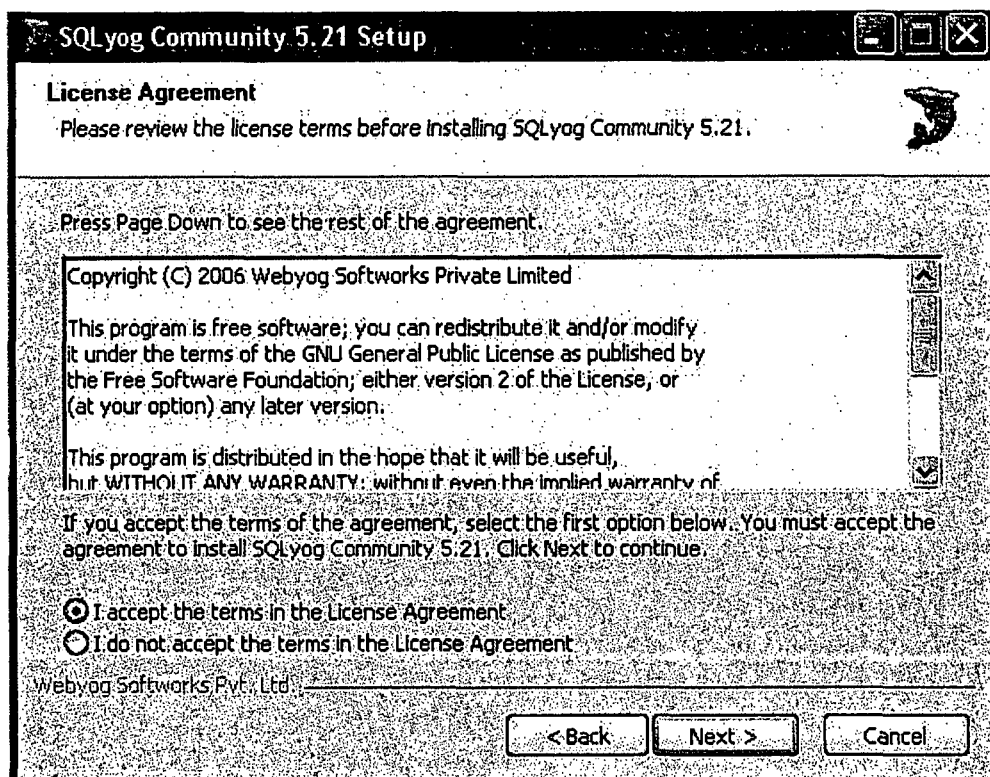
Tras descargar el archivo, lo ejecutaremos y comenzaremos la instalación, siguiendo los siguientes pasos:

1. En la pantalla de bienvenida, pulsaremos "Next".



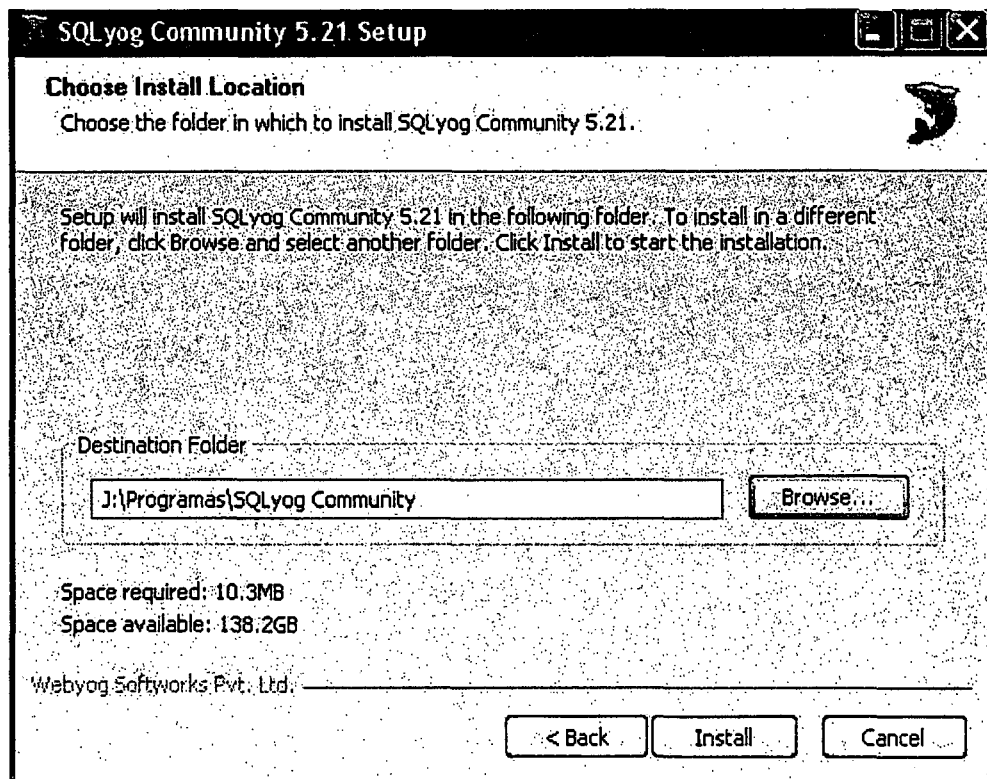
- Figura 107: Instalación de SQLYog 1 -

2. Después nos pedirá que aceptemos la licencia del programa. Debemos marcar la casilla en la que aceptamos y pulsar "Next".



- Figura 108: Instalación de SQLYog 2 -

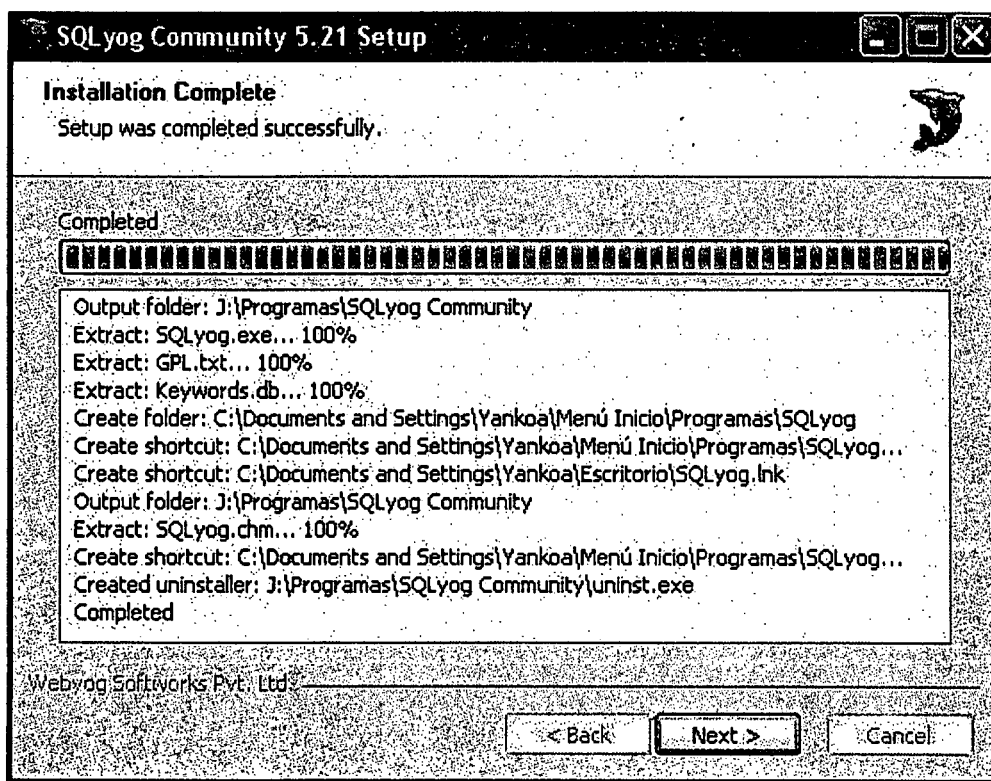
3. Ahora nos pedirá que seleccionamos la ruta en la que deseamos guardar el programa. Se puede dejar la ruta que viene por defecto ó pulsar "Browser" y seleccionar la ruta deseada. Después pulsar "Install".



- Figura 109: Instalación de SQLYog 3 -



4. Cuando la instalación haya sido completada aparecerá la siguiente pantalla, donde tendremos que pulsar "Next".



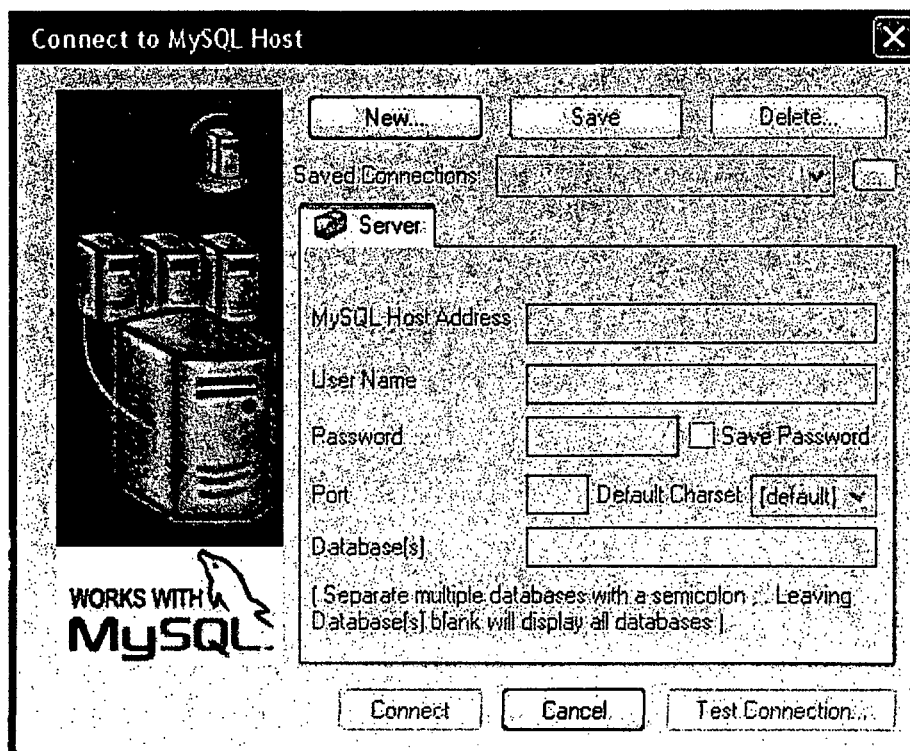
- Figura 110: Instalación de SQLYog 4 -

5. Después nos dará opción a arrancar el programa para configurarlo. Marcaremos la opción y pulsaremos "Finish".



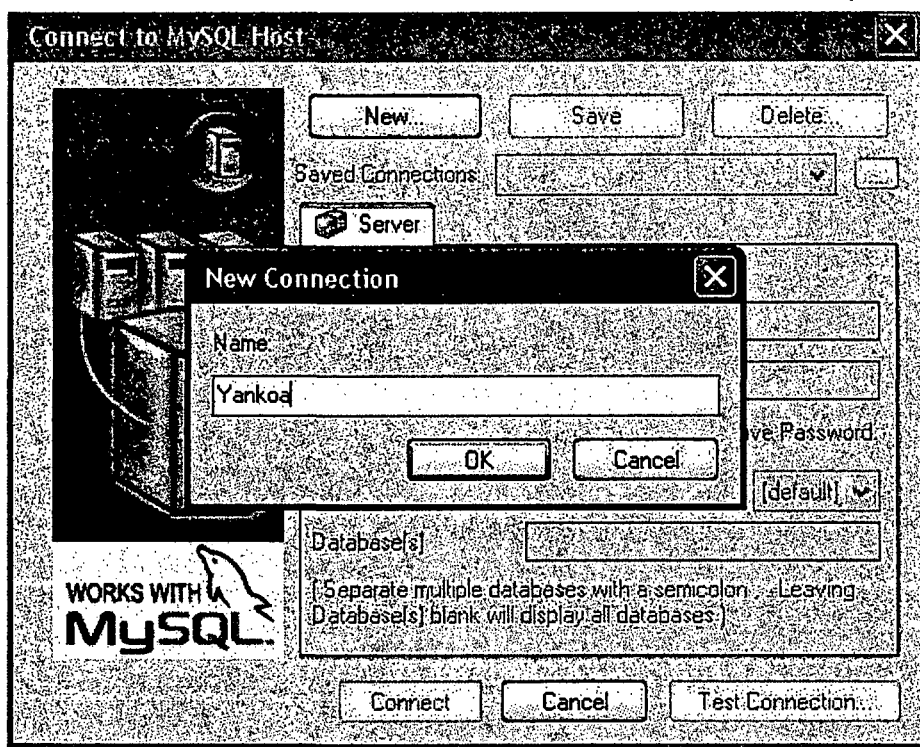
- Figura 111: Instalación de SQLYog 5 -

6. Una vez abierto el programa aparecerá una ventana en el cual deberemos configurar la conexión de la base de datos. Pulsaremos "New..." para crear una nueva conexión.



- Figura 112: Instalación de SQLYog 6 -

7. Pondremos un nombre a la conexión y pulsaremos “OK”.

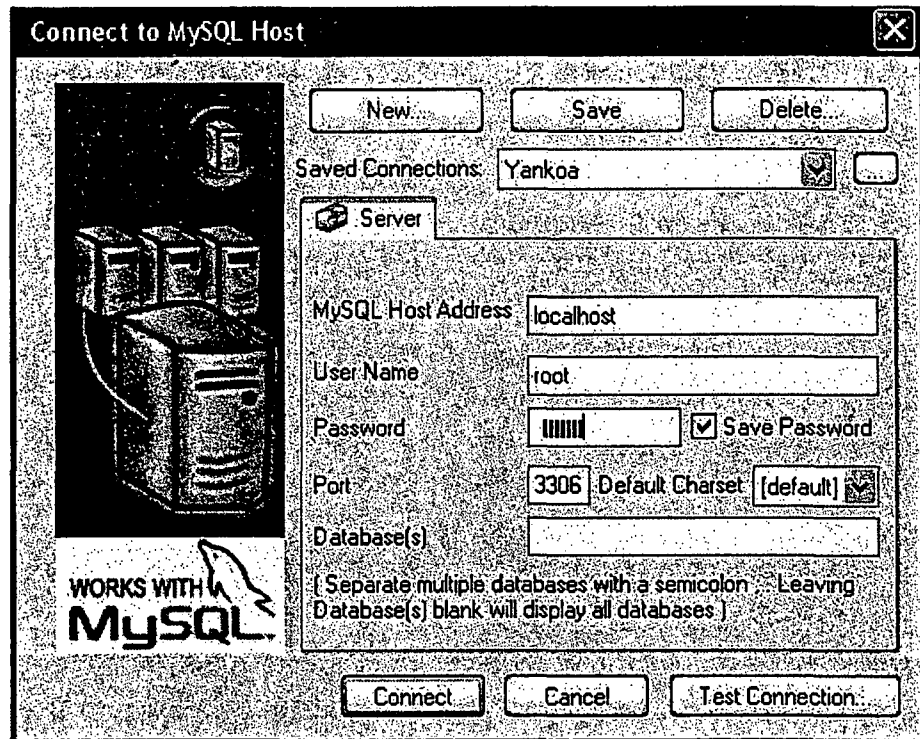


- Figura 113: Instalación de SQLYog 7-

8. Ahora rellenaremos el resto de los datos, de la siguiente manera:

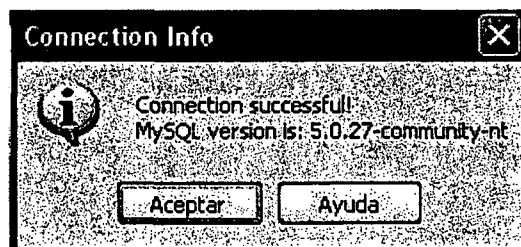
- En el nombre de Host, pondremos “localhost”.
- En el nombre de usuario “root”, ya que somos el administrador de la base de datos.
- En la contraseña la que hayamos elegido durante la instalación de MySQL 5.0.
- El puerto deberá ser el 3306.

Una vez hecho esto pulsaremos sobre el botón “Test Connection...”, para realizar un test a la conexión.



- Figura 114: Instalación de SQLYog 8 -

9. Si el test es correcto aparecerá el siguiente mensaje



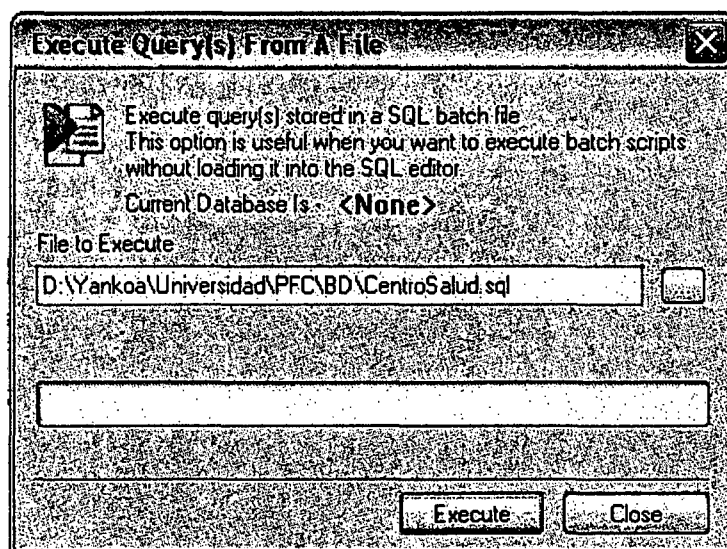
- Figura 115: Instalación de SQLYog 9 -

Pulsaremos “Aceptar” y “Connect” sobre la ventana de conexión.

5.1.4.1.- IMPORTACIÓN DE LA BASE DE DATOS DEL CENTRO DE SALUD

Una vez abierto el programa, será necesario importar el archivo adjunto en el Cd-Room, que contiene las tablas, triggers y datos usados para utilizar la aplicación Web.

En la zona de la izquierda, aparecen algunas bases de datos de ejemplo colgando del servidor “root@localhost”. Debemos pulsar con el botón derecho del ratón sobre el servidor y pulsar sobre la opción “Import from SQL Stament...” apareciendo la siguiente pantalla.



- Figura 116: Importar datos BD -

Aquí debemos seleccionar la ruta en la que se encuentra el fichero de extensión .sql que contiene la base de datos y pulsar "Execute" para que la introduzca en el programa.

Una vez hecho ya tendremos la base datos del centro de salud incorporada.

5.1.5.- INSTALACIÓN DE APACHE TOMCAT 5.5

Para descargar el archivo ejecutable, introduciremos la siguiente dirección <http://archive.apache.org/dist/tomcat/tomcat-5/archive/v5.5.7/bin/> en el navegador. Una vez en la página seleccionaremos el archivo "jakarta-tomcat-5.5.7.exe".

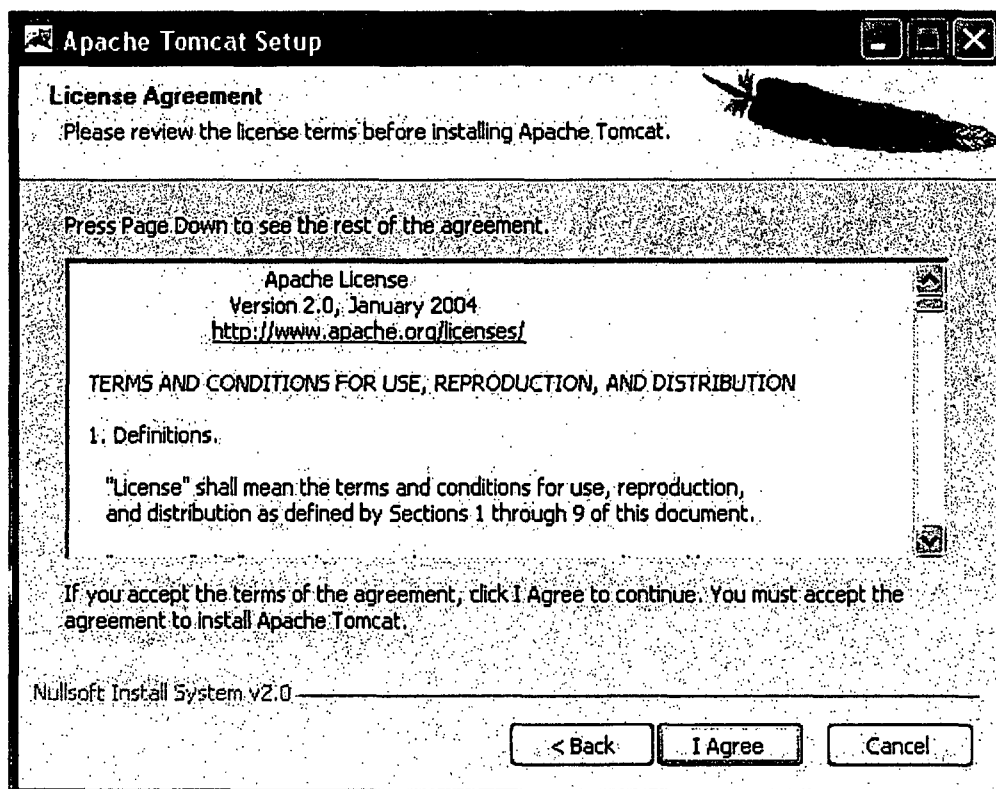
Una vez descargado el archivo pulsaremos sobre y comenzaremos la instalación, siguiendo los siguientes pasos:

1. En la página de bienvenida pulsaremos "Next".



- Figura 117: Instalación de Tomcat 1 -

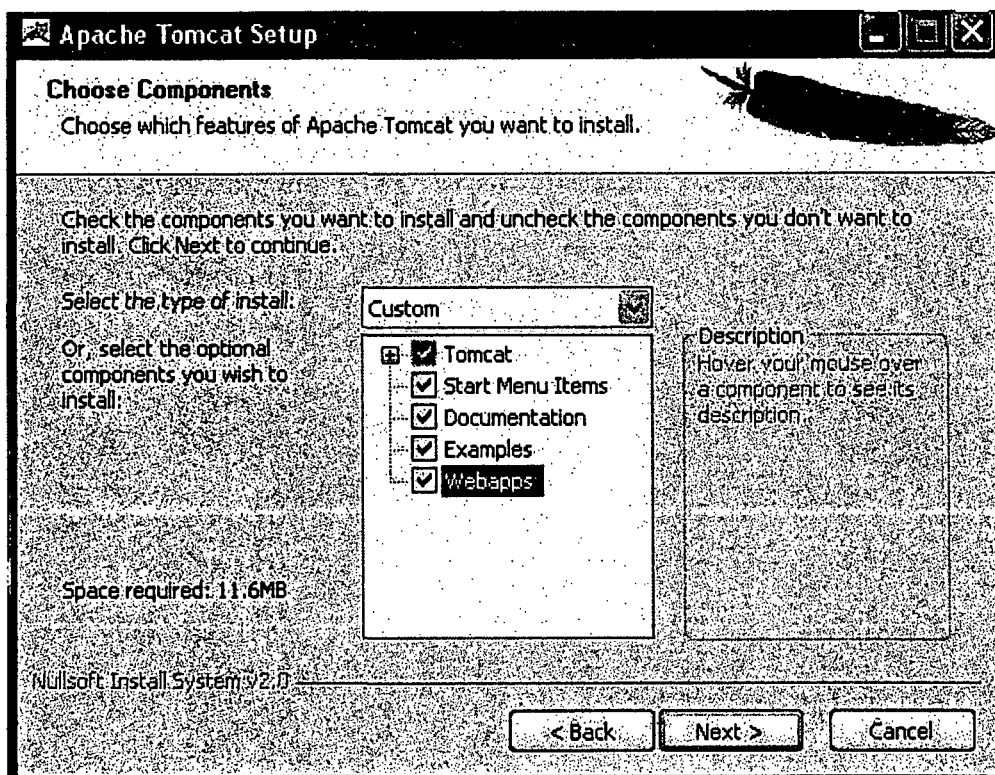
2. Aceptaremos la licencia del programa pulsando "I Agree".



- Figura 118: Instalación de Tomcat 2 -

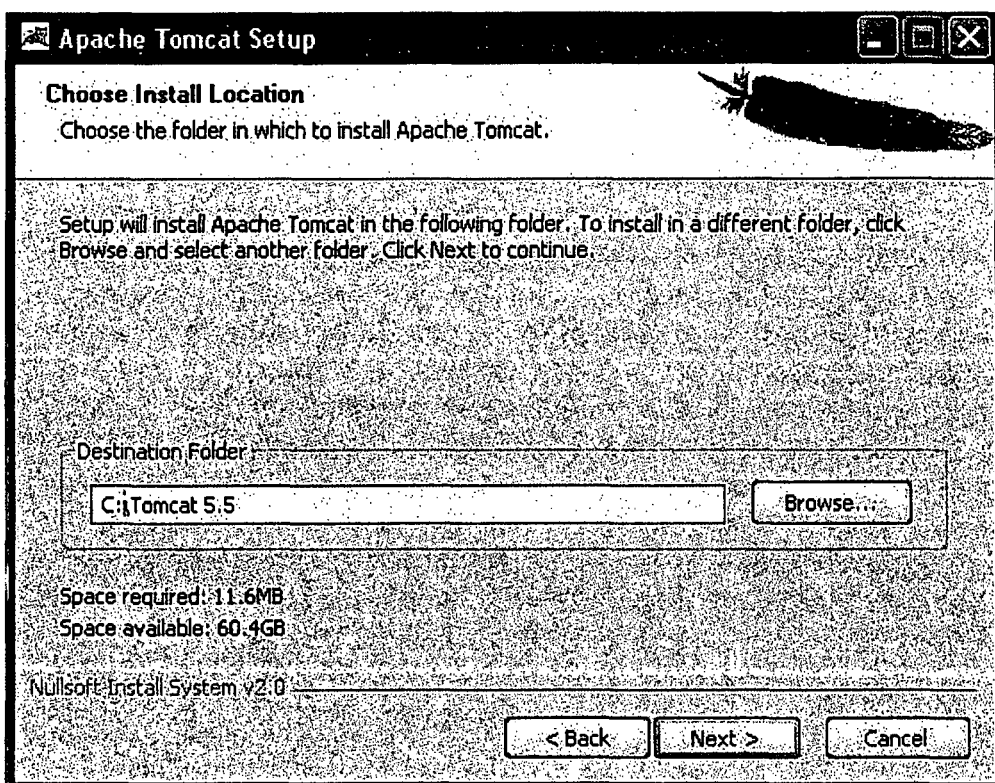


- Después nos pedirá que seleccionemos el tipo de instalación y los componentes que deseamos instalar. Seleccionaremos la instalación propia y todos los componentes. Después pulsaremos "Next".



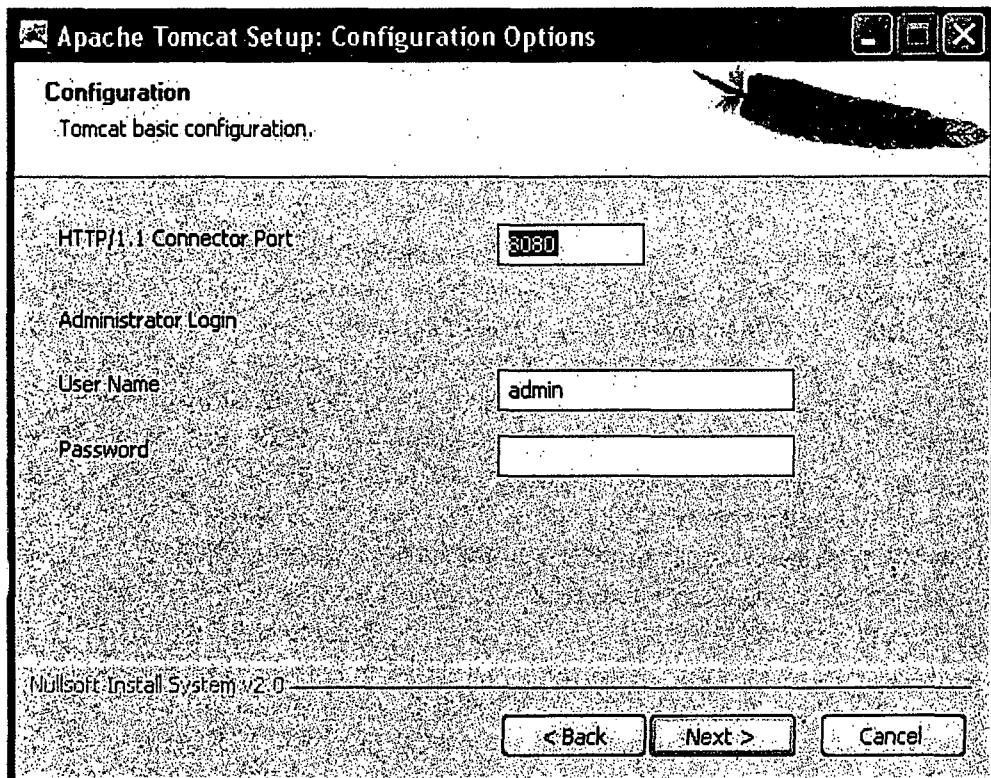
- Figura 119: Instalación de Tomcat 3 -

- Después elegiremos la ruta donde deseemos instalar el programa y pulsaremos "Next".



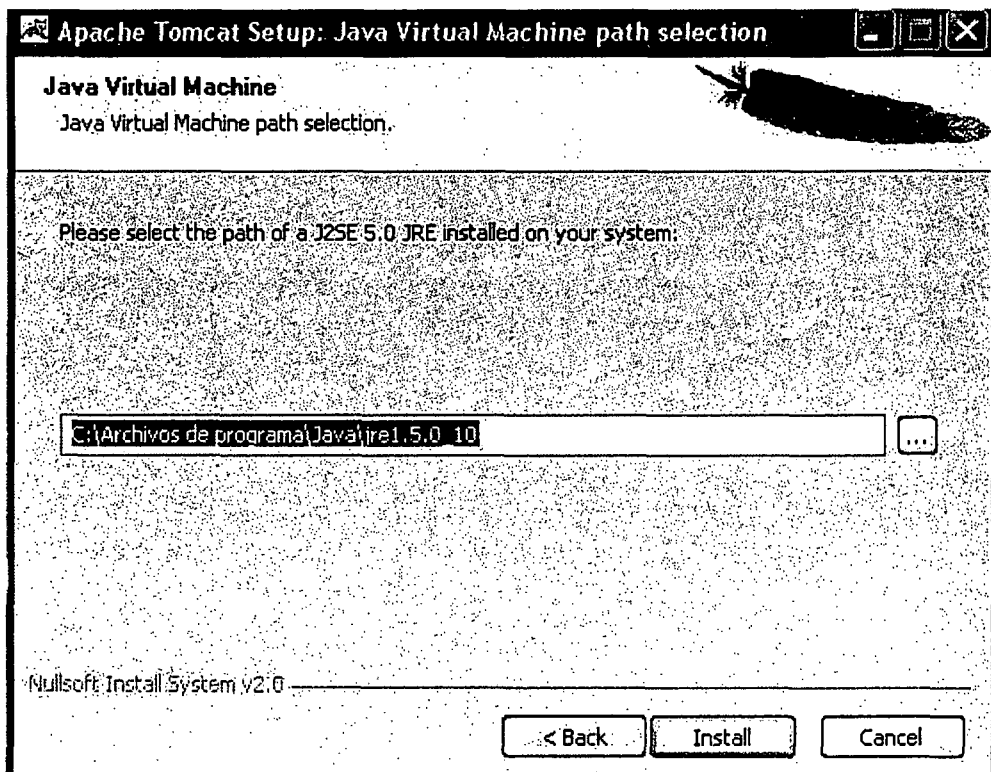
- Figura 120: Instalación de Tomcat 5 -

5. Ahora deberemos introducir los parámetros básicos de configuración de Tomcat.
Primero introduciremos el puerto de Internet por el que correrá Tomcat, el 8080.
Después pondremos el nombre de usuario de Tomcat, por ejemplo, admin.
La contraseña no es necesaria.
Finalmente pulsaremos "Next".



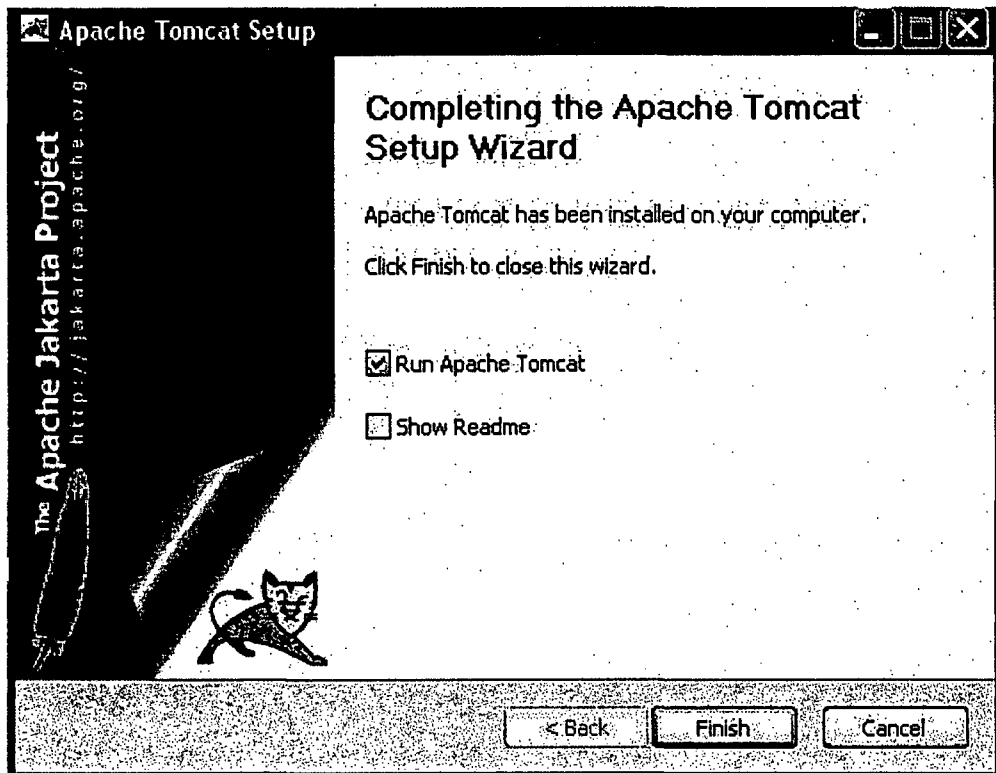
- Figura 121: Instalación de Tomcat 6 -

6. Ahora nos pedirá que introduzcamos la ruta en la que se encuentra en nuestro equipo la Máquina Virtual de Java (JRE). Debemos introducir la ruta en la que la instalamos y pulsar "Install".



- Figura 122: Instalación de Tomcat 7 -

7. Cuando se haya completado la instalación aparecerá la siguiente pantalla. En ella podremos seleccionar ejecutar Tomcat, de forma que comprobemos si realmente funciona, o no hacer nada. De cualquier forma si seleccionamos esta opción, después debemos parar Tomcat, ya que este se activará a través de la herramienta Eclipse WTP. Finalmente pulsaremos "Finish".



- Figura 123: Instalación de Tomcat 8 -

5.1.6.- INSTALACIÓN DE ECLIPSE WTP 1.5.2

Descargar el fichero ejecutable introduciendo la siguiente dirección <http://download.eclipse.org/webtools/downloads/drops/R1.5/R-1.5.2-200610261841/> en un navegador. Una vez en la página, descargar el ejecutable de "WebTools Platform; All-in-one" para la plataforma Windows.

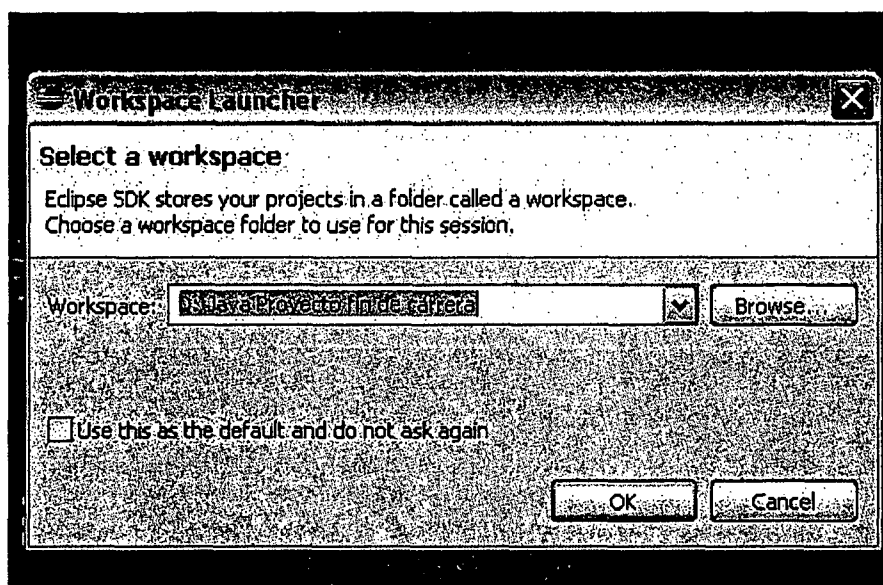
Tras descargar el archivo, descomprimirlo con el descompresor "ZipGenius", de forma que aparecerá una carpeta con el nombre de eclipse. Dentro de la carpeta aparecerá un archivo "eclipse.exe", pulsamos sobre él se abrirá el programa.

5.1.6.1.- CONFIGURACIÓN

Para poder utilizar la aplicación, es necesario configurar Eclipse de la manera correcta.

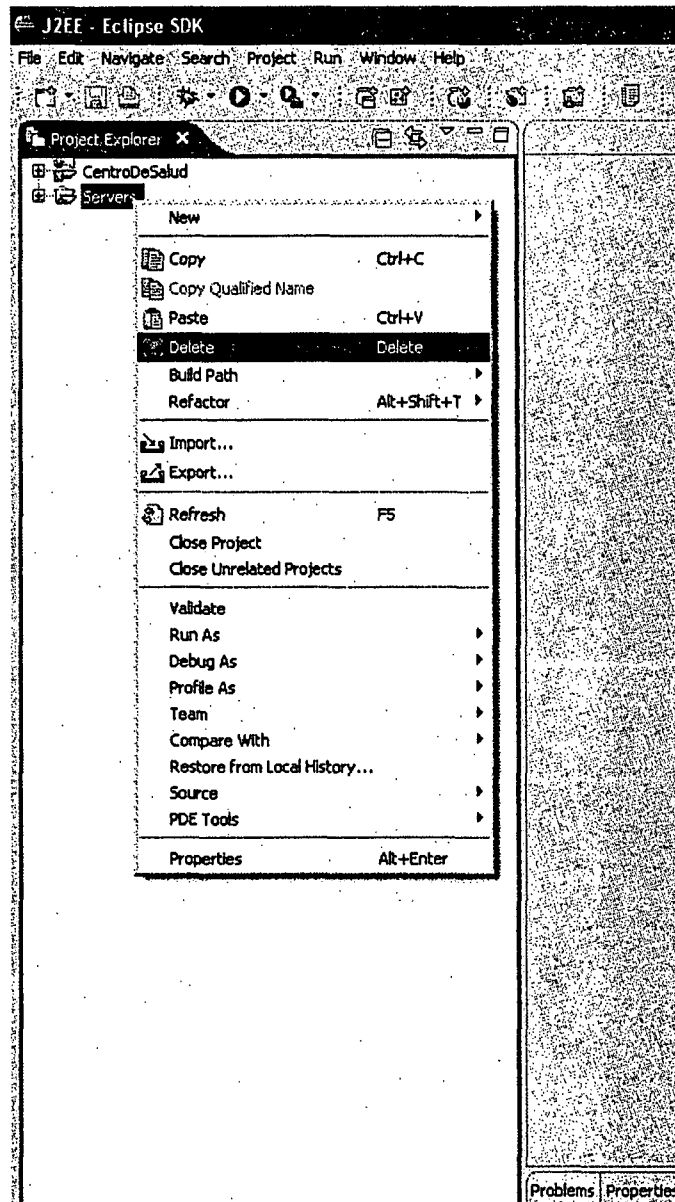
Dado que la aplicación vendrá configurada, será necesario y muy importante seguir los pasos que se van a dar a continuación para modificar la configuración actual.

1. Al abrir Eclipse WTP, aparecerá una ventana en la que se debe indicar el espacio de trabajo. En este caso se debe seleccionar la ruta del Cd-Room en la que se recuenta la carpeta “Java Proyecto fin de carrera” y pulsar “OK”.



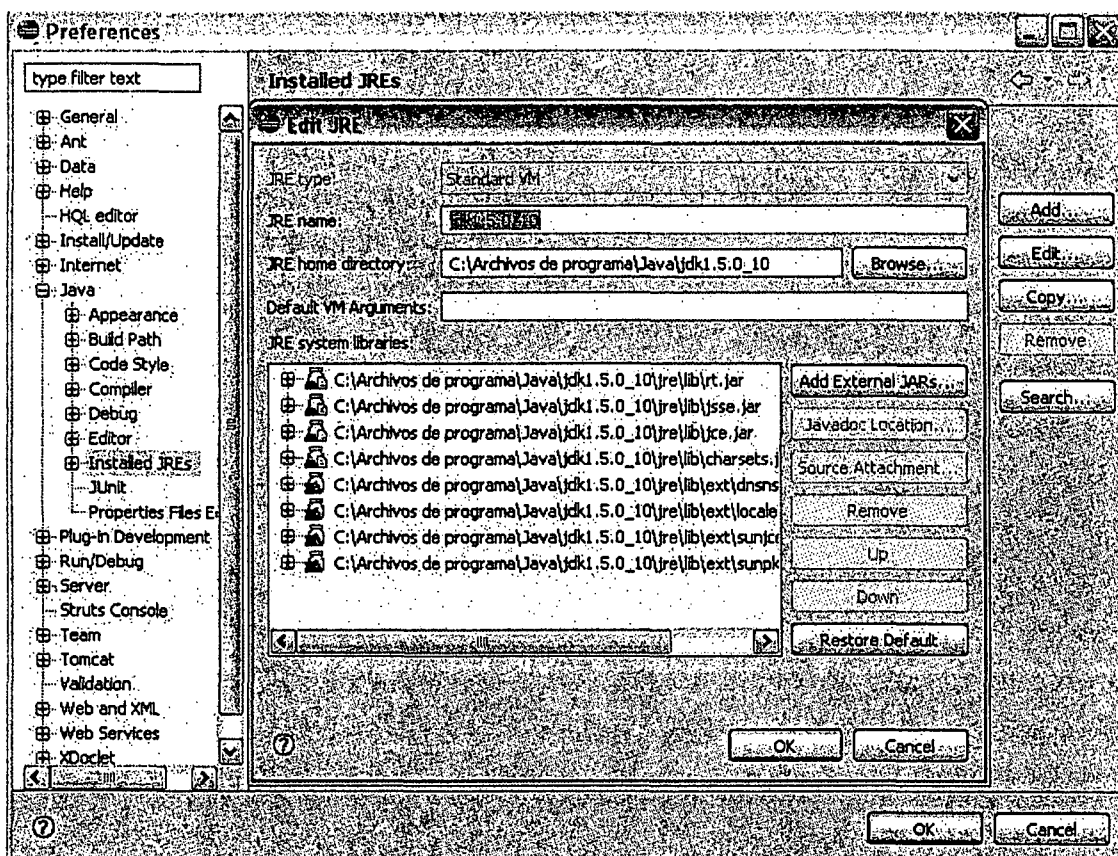
- Figura 124: Configuración de Eclipse WTP 1 -

2. Una vez abierto el programa aparecerá en la ventana de la izquierda el explorador de proyectos, con el proyecto “CentroDeSalud” y “Servers”. Aparecerán muchos errores, pero no debemos alarmarnos ya que son debidos a que no encuentra las rutas de algunos archivos necesarios para su funcionamiento. Debemos eliminar la carpeta “Server” y todo su contenido, ya que después volveremos a crearla con la configuración nueva. Para ello, debemos pulsar sobre la carpeta con el botón derecho del ratón y seleccionar la opción “Delete”, y cuando aparezca una ventana emergente pidiéndonos seleccionar si se desea borrar todo el contenido o solo la carpeta, seleccionaremos la primera opción “Also delete contents under ...”, y aceptamos.



- Figura 125: Borrado de la carpeta Servers -

3. Comenzaremos añadiendo la máquina virtual y el compilador Java, para ello:
 - Pulsamos en Windows / Preferences...
 - Después nos meteremos en Java / Installed JREs.
 - Seleccionaremos la actual Máquina Virtual y pulsaremos sobre "Edit" para modificar su ruta. Apareciendo así la siguiente pantalla.

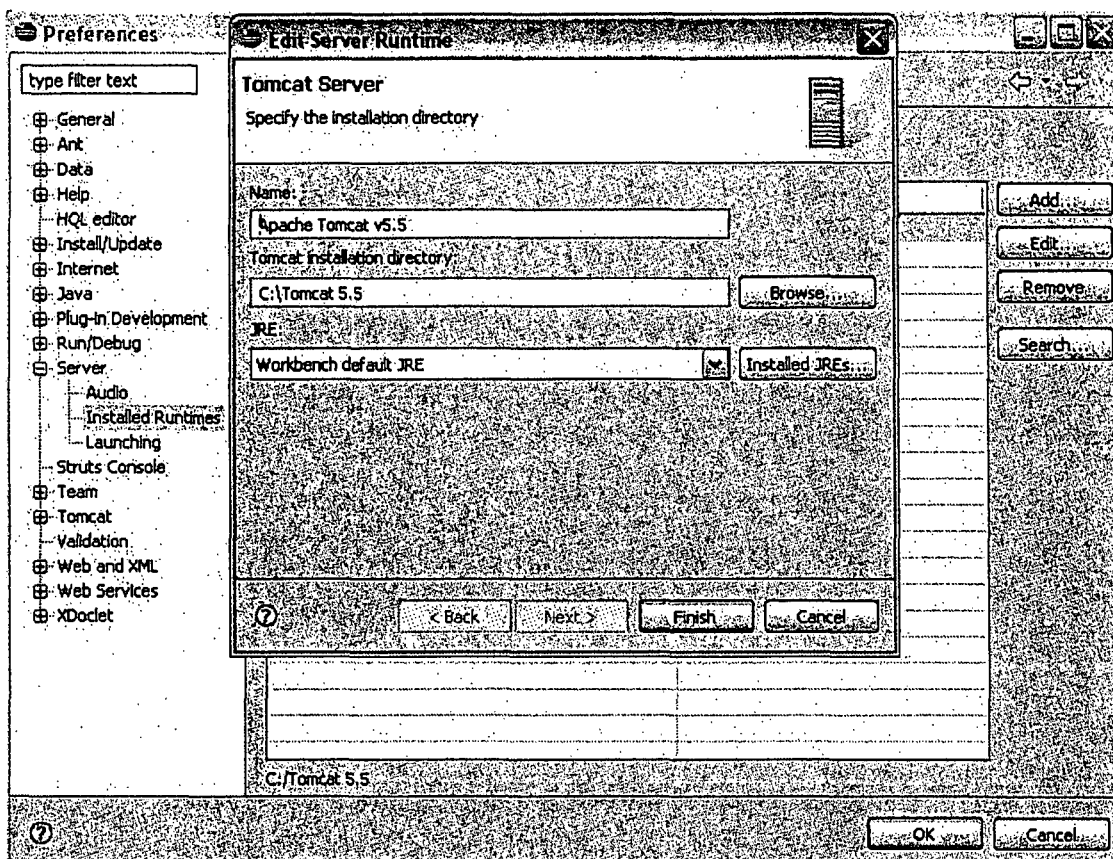


- Figura 126: Configuración de Eclipse WTP 2 -

En “JRE *home directory*” añadiremos la ruta actual donde se encuentra instalada nuestra Máquina Virtual, en este caso JDK, y se instalarán automáticamente todas las librerías necesarias. Después pulsaremos “OK”.

4. Ahora Configuraremos la Máquina Virtual para nuestro servidor, para ello:

- Pulsamos en Windows / Preferences...
- Después nos metemos en Server / Installed Runtimes.
- Señalamos el servidor Apache Tomcat 5.5 y pulsamos en “Edit”, para editarlo, apareciendo así la siguiente pantalla.



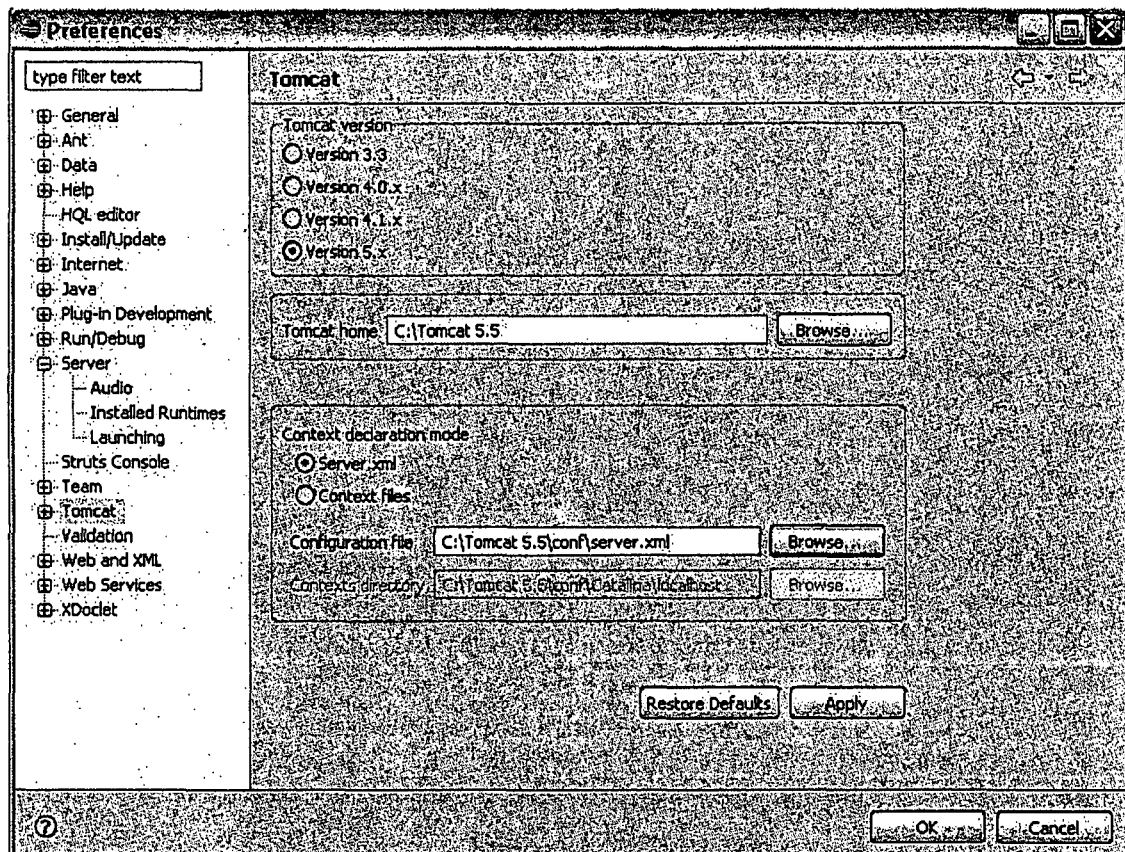
- Figura 127: Configuración de Eclipse WTP 3 -

Debemos pulsar sobre “Browse...” e introducir la ruta donde esta instalado Tomcat en nuestro ordenador y después pulsar “Finish”.

5. Ahora Configuraremos el servidor de aplicaciones, para ello:

- Pulsaremos sobre Windows / Preferences.
- Después nos metemos en Tomcat, seleccionaremos la ruta en la que esta instalado Tomcat en nuestro ordenador y la ruta en la que se encuentra el archivo “Server.xml”.

Atención, en caso de no aparecer dentro de Window / Preferences, Tomcat, omitiremos este paso, ya que después se configurará cuando creemos el servidor.

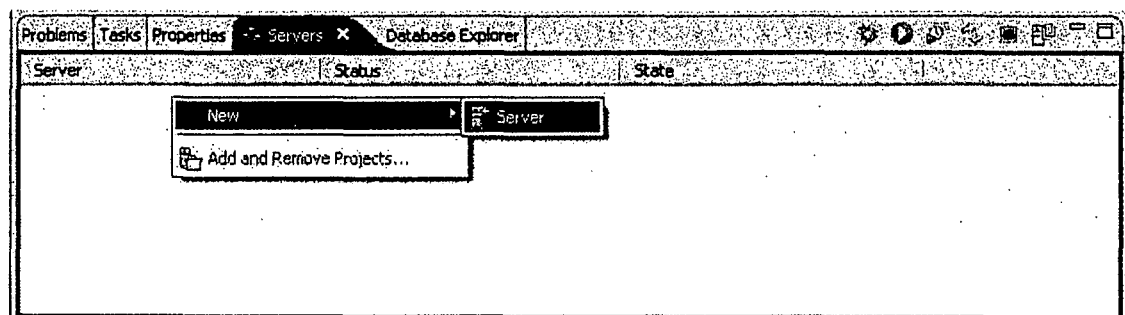


- Figura 128: Configuración de Eclipse WTP 4 -

6. Añadiremos nuestro servidor al proyecto, para ello:

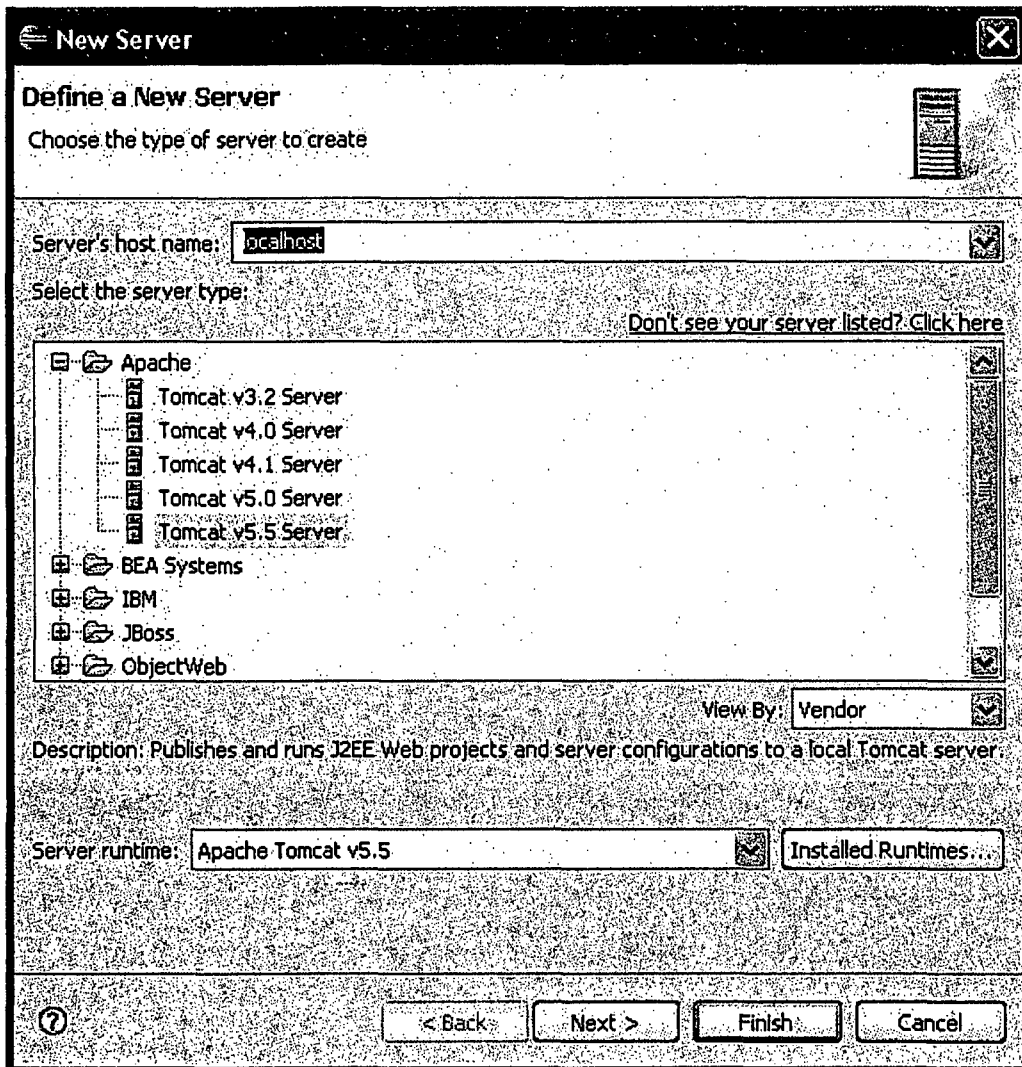
Primeramente, eliminaremos el actual servidor, pulsando con el botón derecho del ratón sobre él, y seleccionando la opción "Delete", dejando por defecto la opción de borrado.

Después, en la ventana Server en la parte inferior pulsaremos con el botón derecho del ratón y elegiremos New / Server.



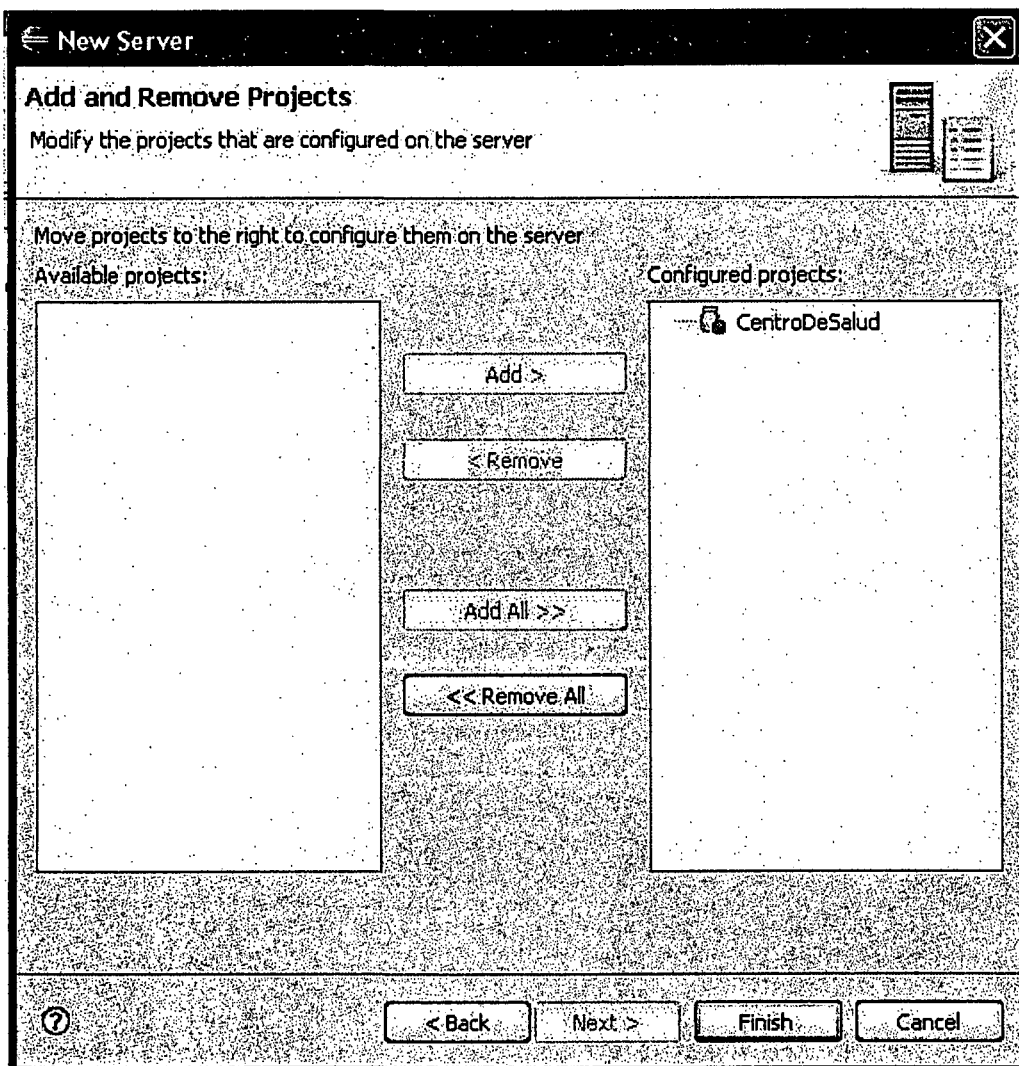
- Figura 129: Configuración de Eclipse WTP 5 -

Después le pondremos un nombre al host del server y seleccionamos nuestro servidor tal y como se muestra en la imagen. Después pulsaremos "Next".



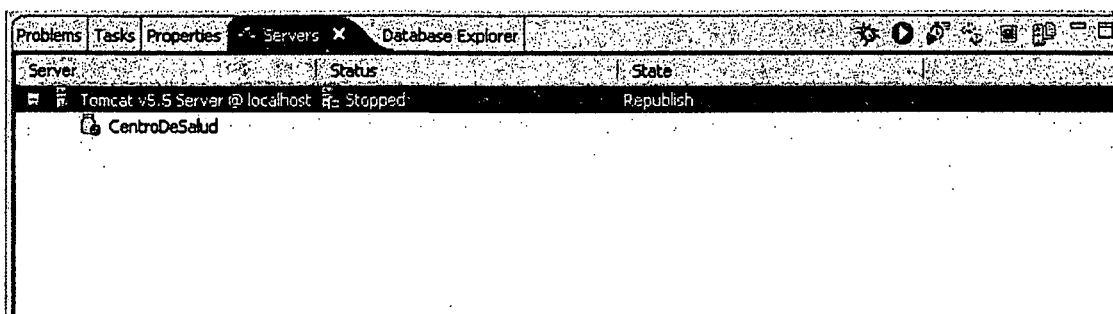
- Figura 130: Configuración de Eclipse WTP 6 -

Añadiremos nuestro proyecto a los configurados para que funcione sobre el servidor.



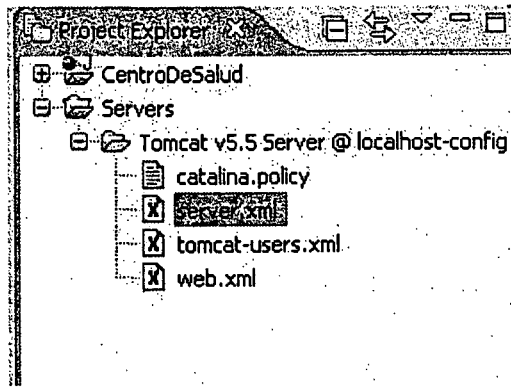
- Figura 131: Configuración de Eclipse WTP 7 -

Tras pulsar sobre "Finish" terminaremos la configuración del servidor, y nuestra pantalla Server tendrá un aspecto como el siguiente.



- Figura 132: Configuración de Eclipse WTP 8 -

Tras estos pasos observaremos también que en el explorador del proyecto ha aparecido una nueva carpeta denominada "Servers".



- Figura 133: Configuración de Eclipse WTP 9 -

Debemos abrir el fichero server.xml y borrar a partir de la etiqueta “<context>” todo, añadiendo en ese lugar lo siguiente:

```
<Context docBase="CentroDeSalud" path="/CentroDeSalud"
reloadable="true" source="org.eclipse.jst.j2ee.server:CentroDeSalud">

<Resource name="jdbc/pfc" auth="Container" type="javax.sql.DataSource"
maxActive="10000" username="root" password="prueba"
driverClassName="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost:3306/insalud">
</Resource>

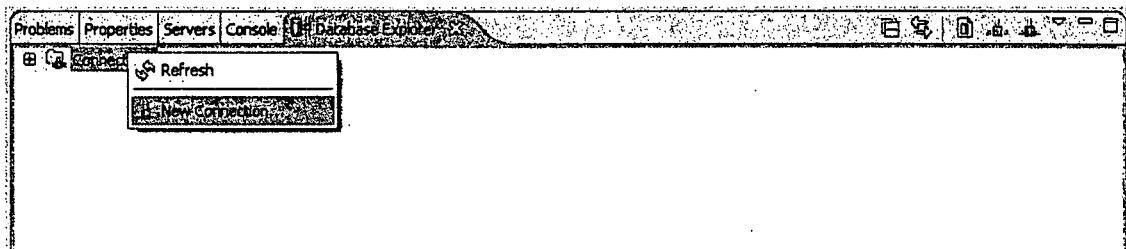
</Context></Host>
</Engine>
</Service>
</Server>
```

Por supuesto la contraseña, debe corresponderse con la de la base de datos.

7. Opcionalmente podremos añadir una vista de la base de datos a Eclipse, de esta forma nos aseguramos que existe una conectividad entre la base de datos y Eclipse; para ello en la ventana “Database Explorer” en la parte inferior del programa, realizamos los siguientes pasos:

Primeramente eliminaremos la conexión existente pulsando sobre ella con el botón derecho del ratón y seleccionando la opción “Delete”.

Después creamos una nueva conexión.

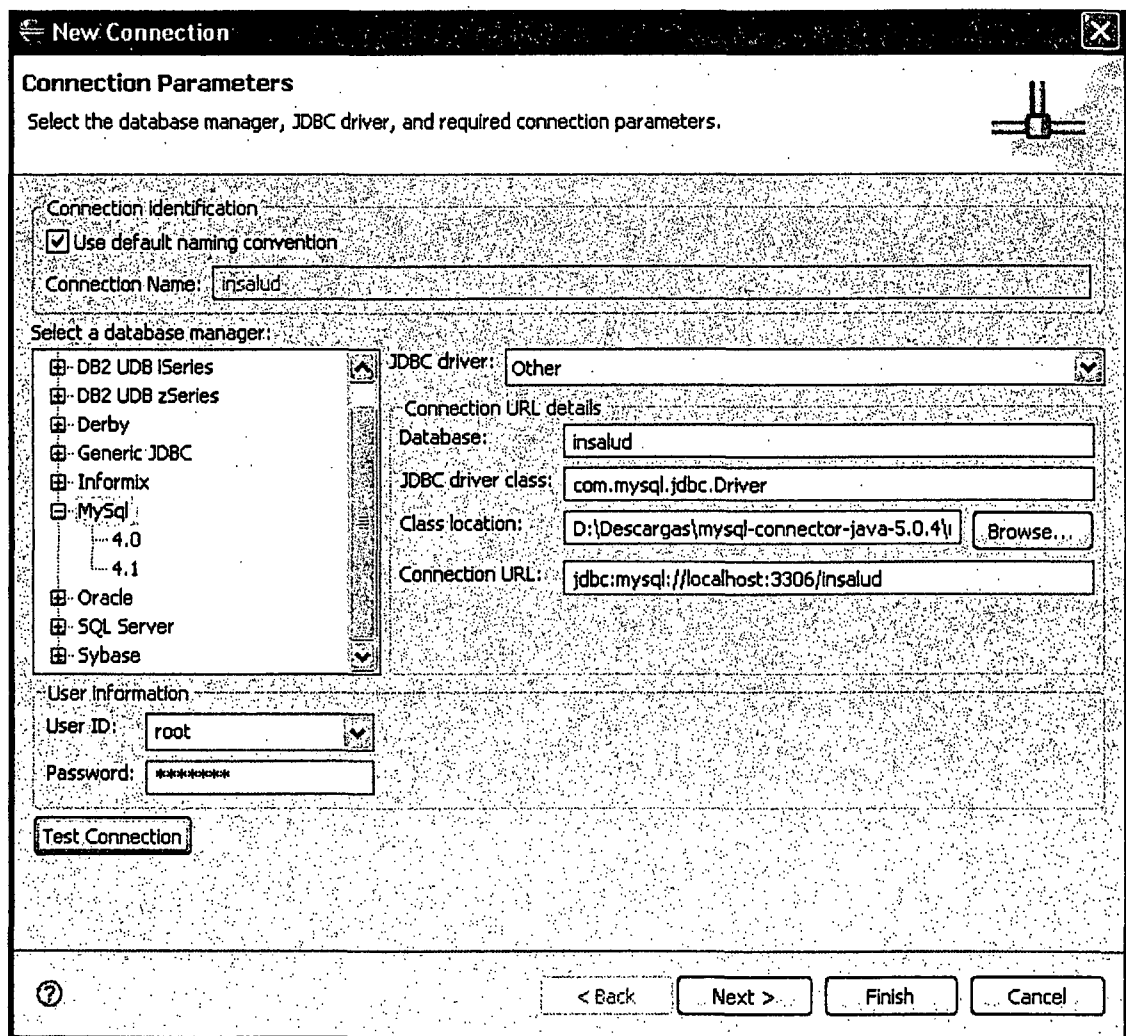


- Figura 134: Configuración de Eclipse WTP 10 -



Señalamos nuestro proveedor de BD, en este caso ponemos la máxima versión de MySQL aunque no es la que estamos utilizando, ya que Eclipse no tiene la versión 5.0.

- Introducimos el nombre de la BD, "insalud".
- Introducimos el nombre del driver de la BD, "com.mysql.jdbc.Driver".
- Introducimos la ruta en la que tenemos descomprimido dicho driver.
- Introducimos la ruta de conexión, "jdbc:mysql://localhost:3306/insalud".
- Finalmente introducimos el nombre de usuario y contraseña de la base de datos que hemos puesto cuando hemos instalado MySQL 5.0 y hacemos un test de conexión. Si todo es correcto, pulsamos en "Finish" y ya habremos comprobado que existe conexión entre Eclipse y MySQL. Es recomendable antes de realizar el test de conexión, desactivar cualquier tipo de firewall, que se tenga instalado en la máquina.



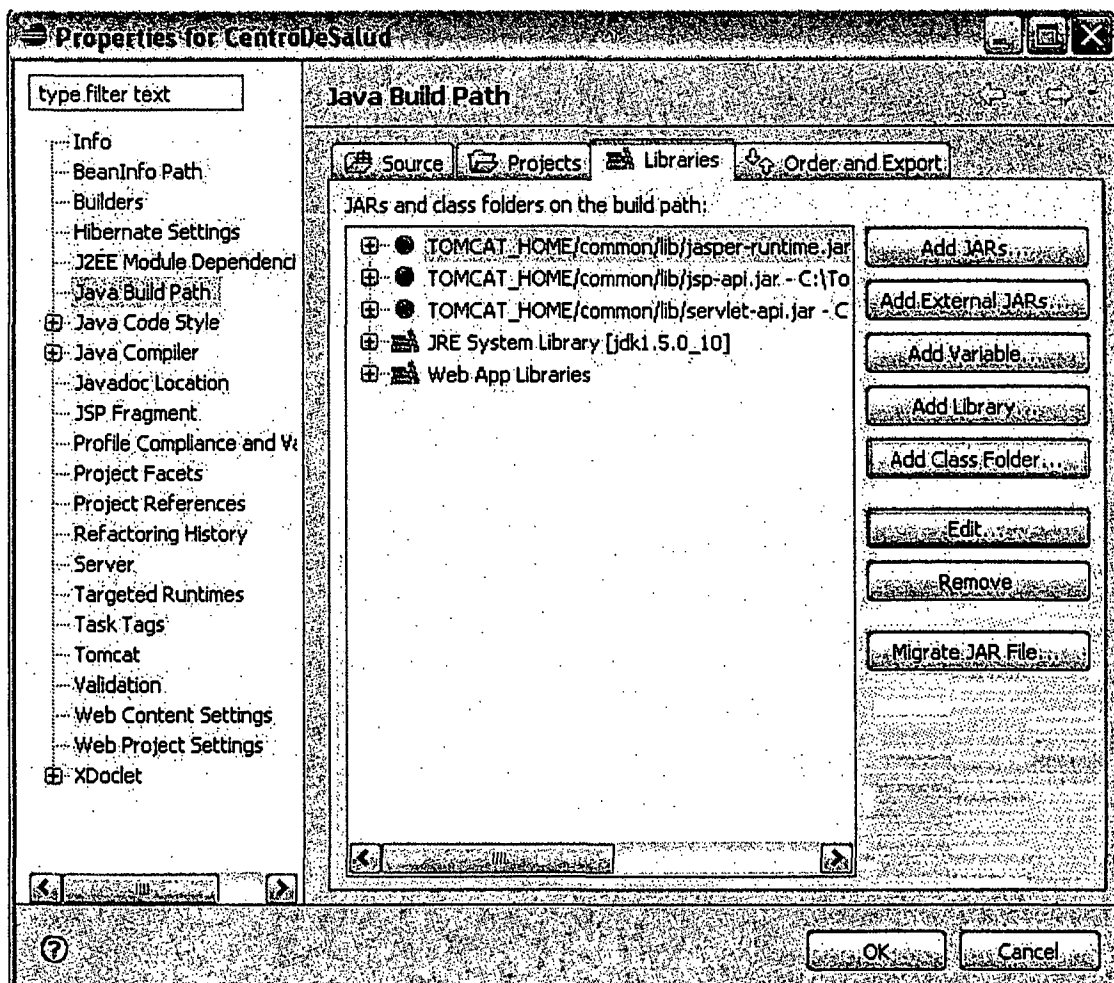
- Figura 135: Configuración de Eclipse WTP 11 -

8. Ahora introduciremos una copia del driver de la base de datos en el servidor de aplicaciones, Tomcat 5.5.



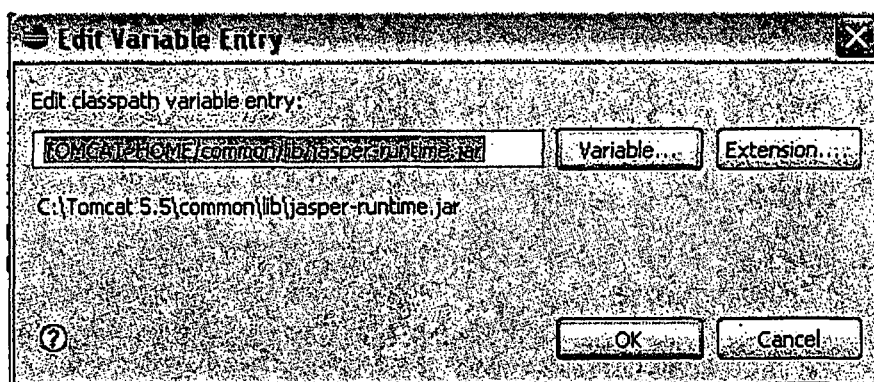
Para ello deberemos abrir la carpeta en la que tenemos Tomcat, e ir navegando hasta "C:\Tomcat 5.5\common\lib". Dentro de la carpeta "lib", pegar el driver de la base de datos.

9. Finalmente pulsaremos con el botón derecho sobre el proyecto y seleccionaremos "Properties". Dentro de properties deberemos ir a la opción Java Build Path / Libraries.



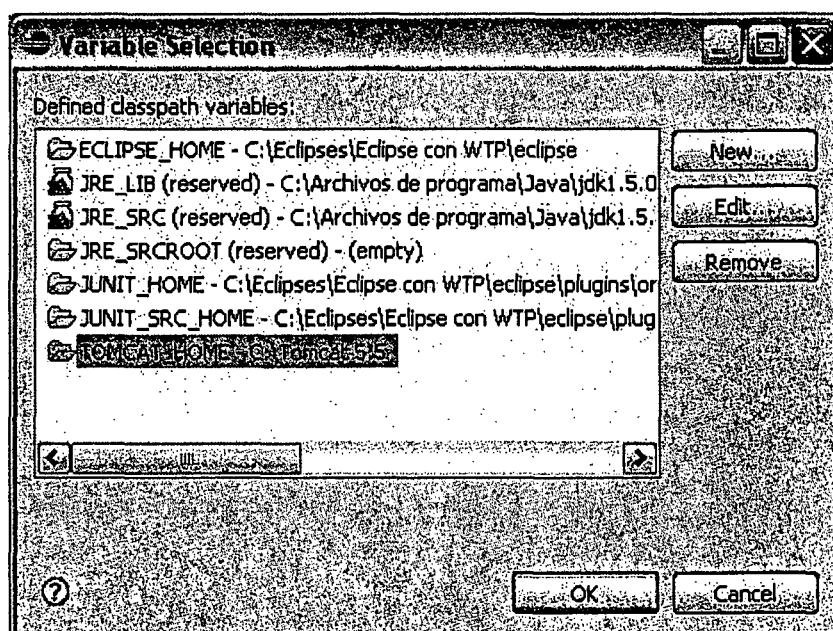
- Figura 136: Configuración de Eclipse WTP 12 -

Una vez allí pulsaremos sobre cualquiera de las librerías de TOMCAT_HOME, y daremos a "Edit".



- Figura 137: Configuración de Eclipse WTP 13 -

Pulsaremos sobre "Variable...".




- Figura 138: Configuración de Eclipse WTP 14 -

Finalmente editaremos la variable de entorno de Tomcat, poniendo la ruta donde se encuentra instalado actualmente Tomcat.

Para asegurarnos de que la nueva configuración tiene efecto, se recomienda cerrar el programa y volver a abrirlo. En caso de seguir apareciendo errores, volver a realizar el paso 9.

Tras esto habremos finalizado la configuración de Eclipse WTP para el proyecto.

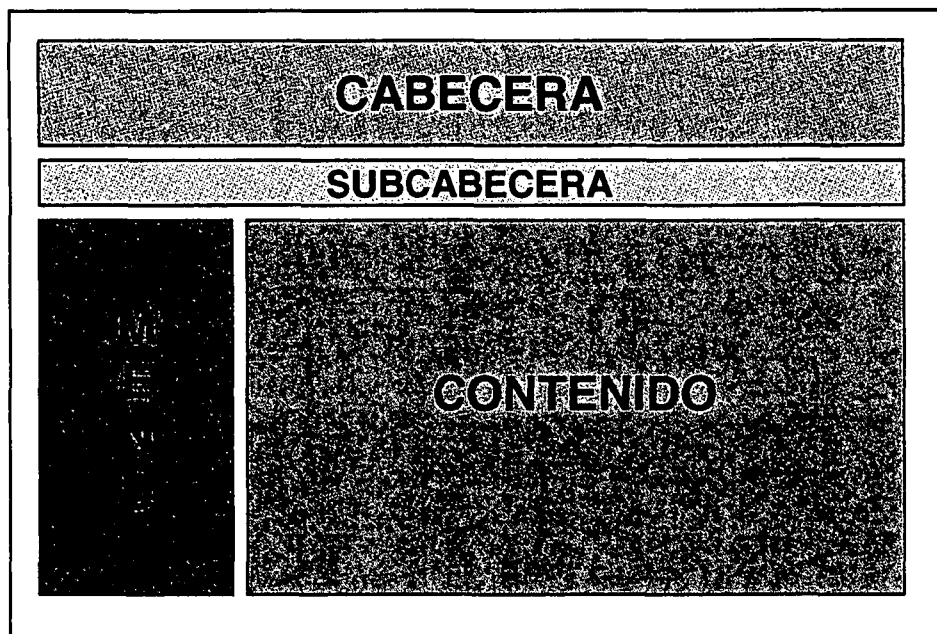
Solo nos faltará encender el servidor de aplicaciones pulsando sobre el botón  de la pestaña "Servers" en la parte inferior de Eclipse, para poder comenzar a probar la aplicación.

5.2.- MANUAL DE USUARIO

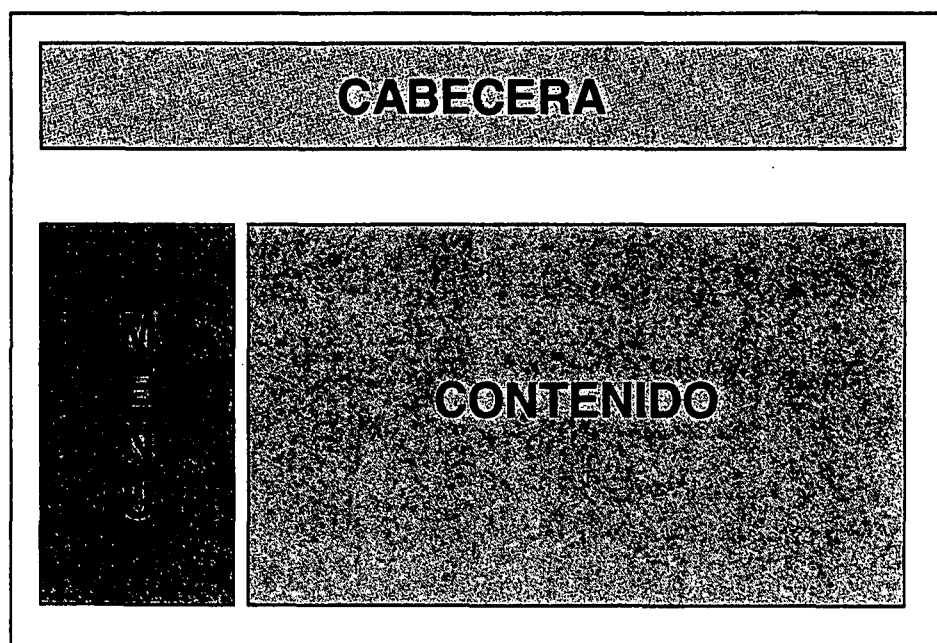
5.2.1.- ESTRUCTURA DE LOS INTERFACES

Existen tres interfaces en la aplicación, los cuales se diferencian por el número de capas por las que están formados.

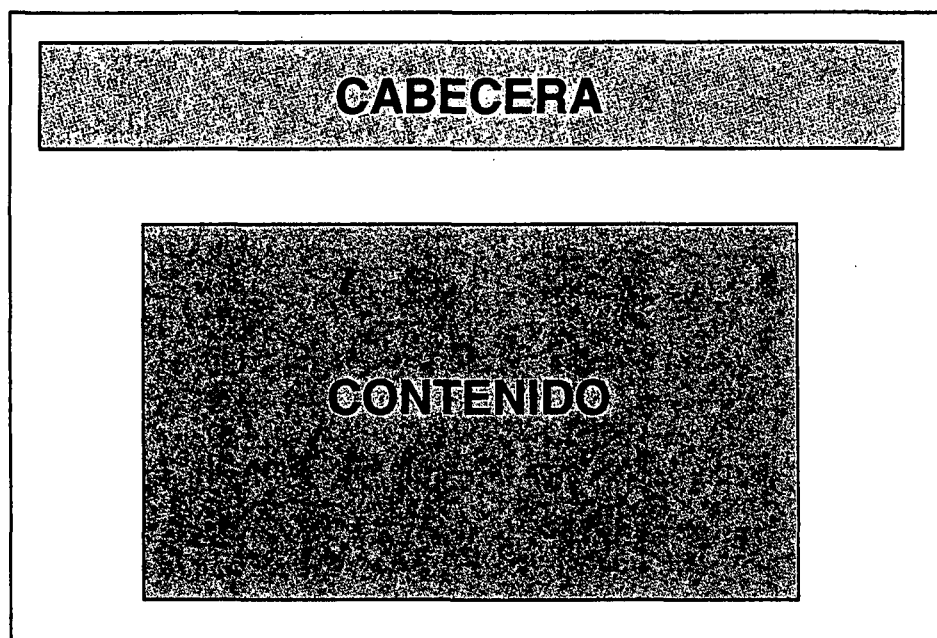
En las siguientes imágenes, se distingue cada interfaz.



- Figura 139: Configuración Interfaz del administrador-



- Figura 140: Interfaz del médico-



- Figura 141: Interfaz del usuario-

Los elementos que componen las interfaces son:

- **Cabecera:** En esta parte se encuentra el logo de la sanidad madrileña y el nombre del centro de salud.
- **Subcabecera:** Este elemento contiene el tipo de usuario que se encuentra logado en la aplicación (administrador ó médico).
- **Menu:** Contiene todos los enlaces a las funcionalidades que se pueden realizar.
- **Contenido:** En esta parte se sitúan los contenidos y datos necesarios para cada una de las funcionalidades a la cuales se vaya accediendo.

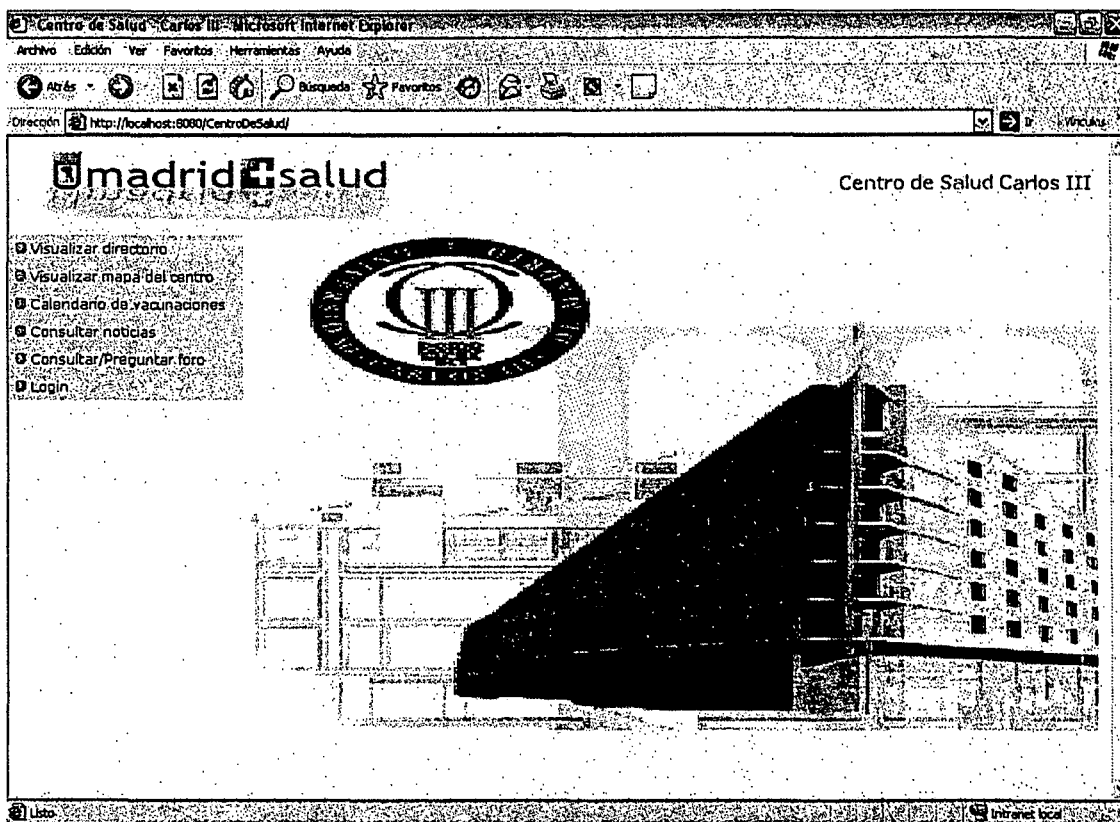
5.2.2.- ¿CÓMO EMPEZAR?

Para acceder a la aplicación es necesario introducir el navegador Microsoft Internet Explorer 6, la siguiente dirección:

http://<host>:<puerto>/CentroDeSalud/

Donde “host” se corresponderá con el nombre de una máquina y “puerto”, el puerto por el que correrá la aplicación.

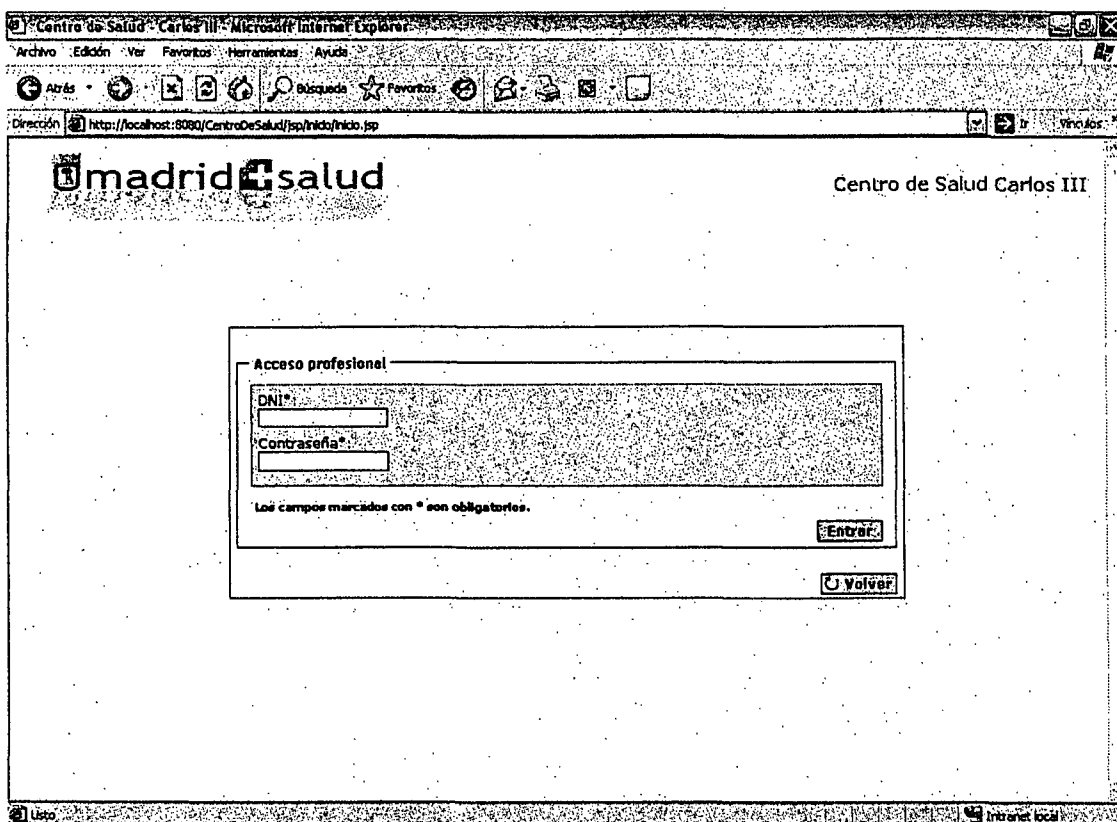
Una vez introducida aparecerán todas las opciones que puede realizar cualquier usuario de la aplicación sin necesidad de registrarse como administrador ó médico.



- Figura 142: Bienvenida -

5.2.3.- LOGEARSE COMO ADMINISTRADOR Ó MÉDICO

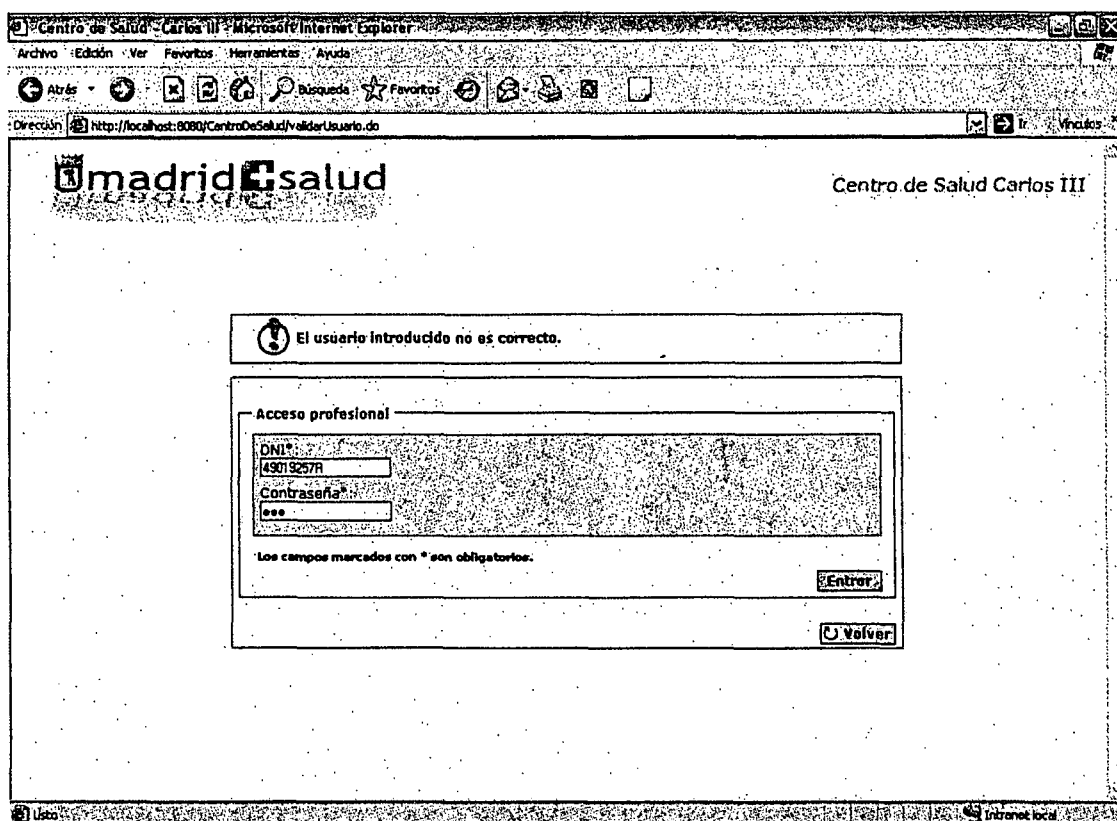
Tras entrar en la aplicación si eres administrador ó médico, opcionalmente puedes pulsar sobre la opción del menú "Login", pareciendo la siguiente pantalla.



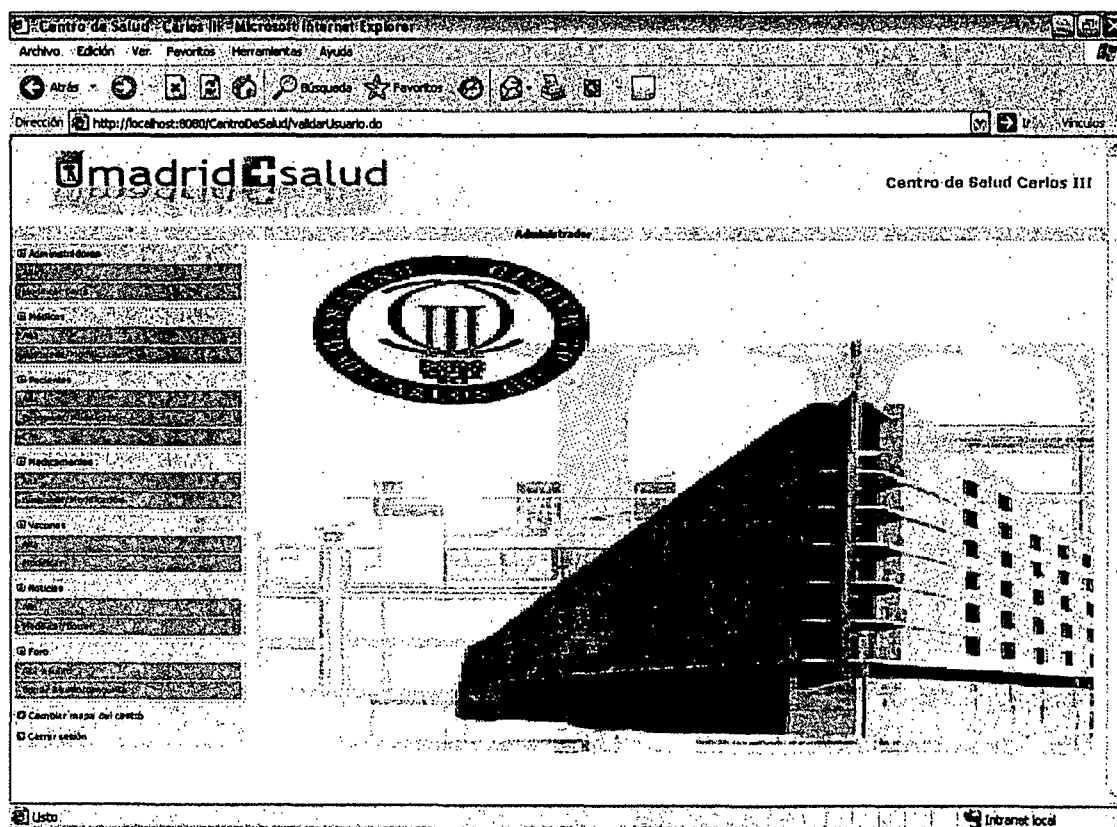
- Figura 143: Acceso profesional -

En esta pantalla el administrador ó médico deberán introducir su DNI y contraseña, la cual habrá sido previamente suministrada por un administrador del centro de salud, para poder acceder a la aplicación. También tienen opción de regresar a la pantalla anterior pulsando sobre el botón “Volver”

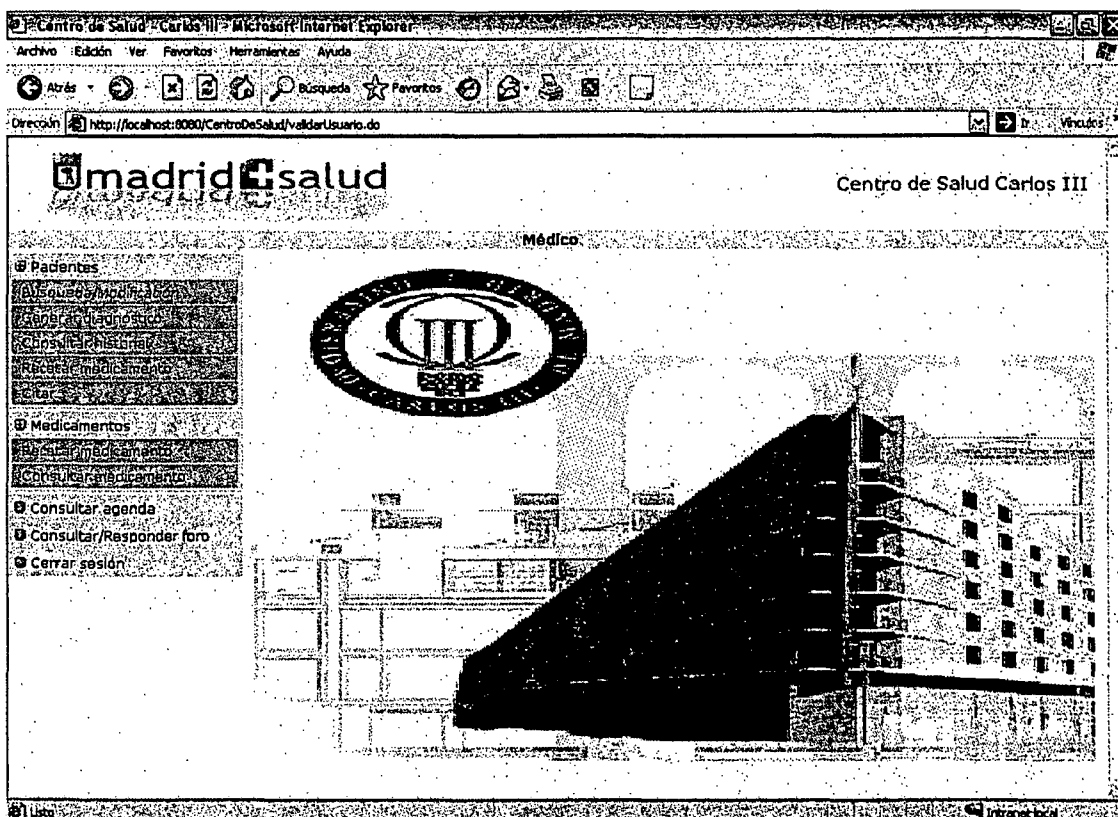
Tras introducir los datos, se comprobará si estos son correctos. En caso de no serlos se avisará de ello, dando la oportunidad de volver a introducir los datos. En caso de no producirse ningún error se el administrador ó médico, será redirigido a la pantalla de bienvenida con sus opciones correspondientes en el menú.



- Figura 144: Acceso incorrecto -



- Figura 145: Bienvenida administrador -



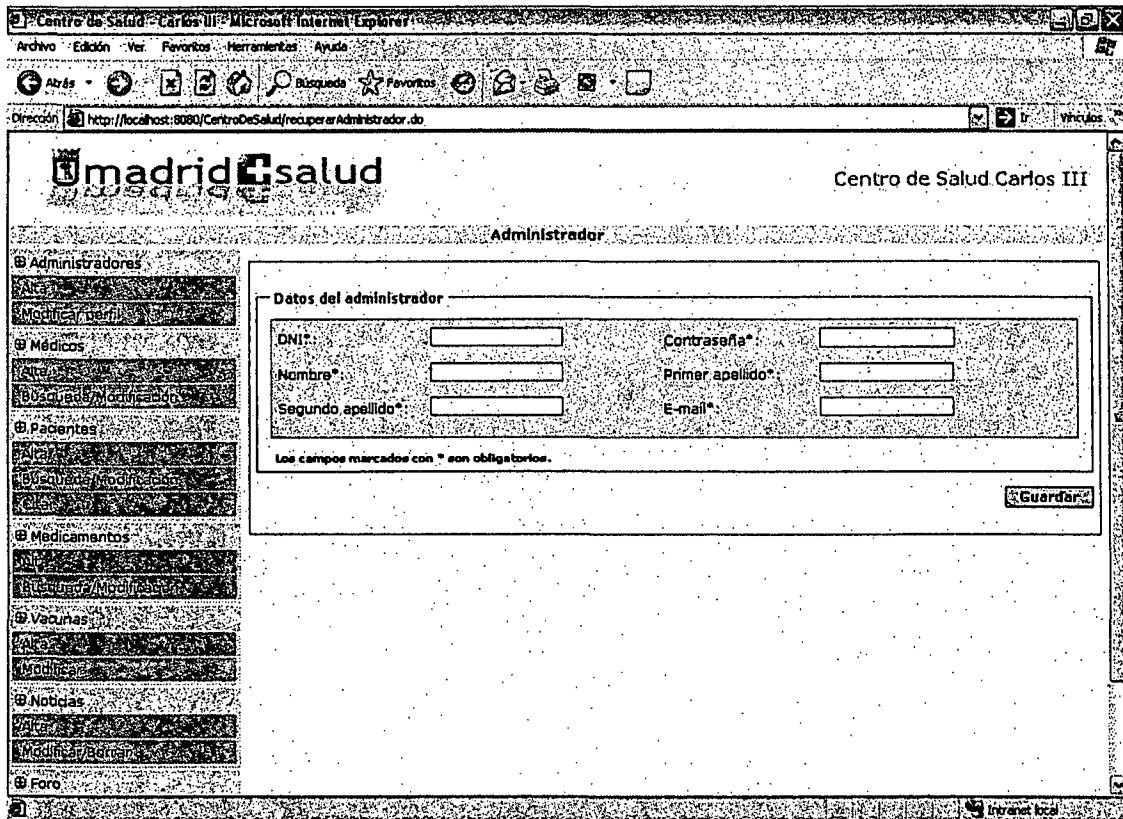
- Figura 146: Bienvenida médico -

5.2.4.- OPCIONES DEL ADMINISTRADOR

Para acceder a dichas opciones un administrador del centro debe haberse registrado correctamente en la aplicación. Partiendo de la pantalla de bienvenida, podrá realizar las acciones que aparecen en el menú de la derecha.

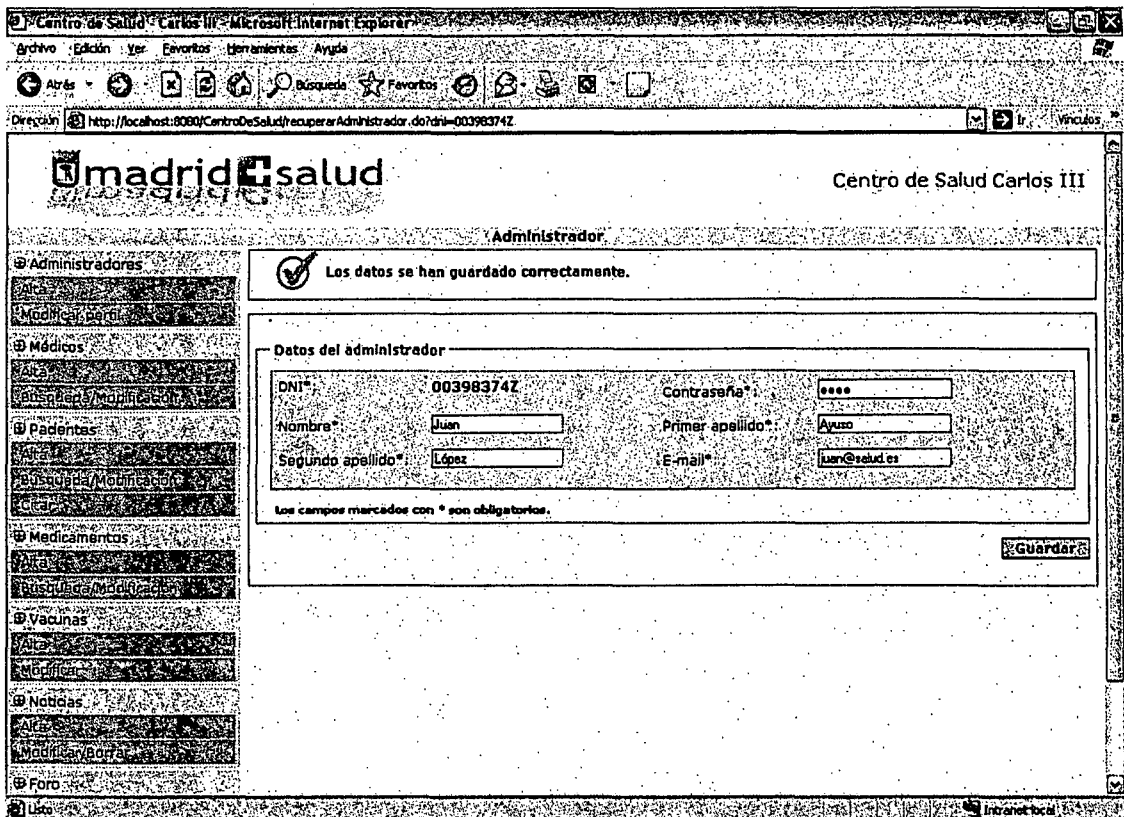
5.2.4.1.- DAR DE ALTA UN NUEVO ADMINISTRADOR

Para dar de alta un nuevo administrador, el administrador debe primeramente pulsar en la opción del menú "Alta" (Administradores). De esta forma le aparecerá una pantalla en la que se le pedirá que introduzca los datos personales del administrador que se va dar de alta.



- Figura 147: Alta administrador-

Una vez introduzca los campos necesarios, el administrador pulsará en guardar. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo en los datos introducidos, el administrador podrá modificarlos y tras pulsar de nuevo en guardar estos se actualizarán.



Centro de Salud Carlos III - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Alrés

Dirección http://localhost:8080/CentroDeSalud/recuperarAdministrador.do?dni=00398374Z

Madrid Salud

Centro de Salud Carlos III

Administrador

Los datos se han guardado correctamente.

Datos del administrador

DNI* 00398374Z Contraseña* ****

Nombre* Juan Primer apellido* Ayuso

Segundo apellido* López E-mail* juan@salud.es

Los campos marcados con * son obligatorios.

Guardar

Administradores

Médicos

Pacientes

Medicamentos

Vacunas

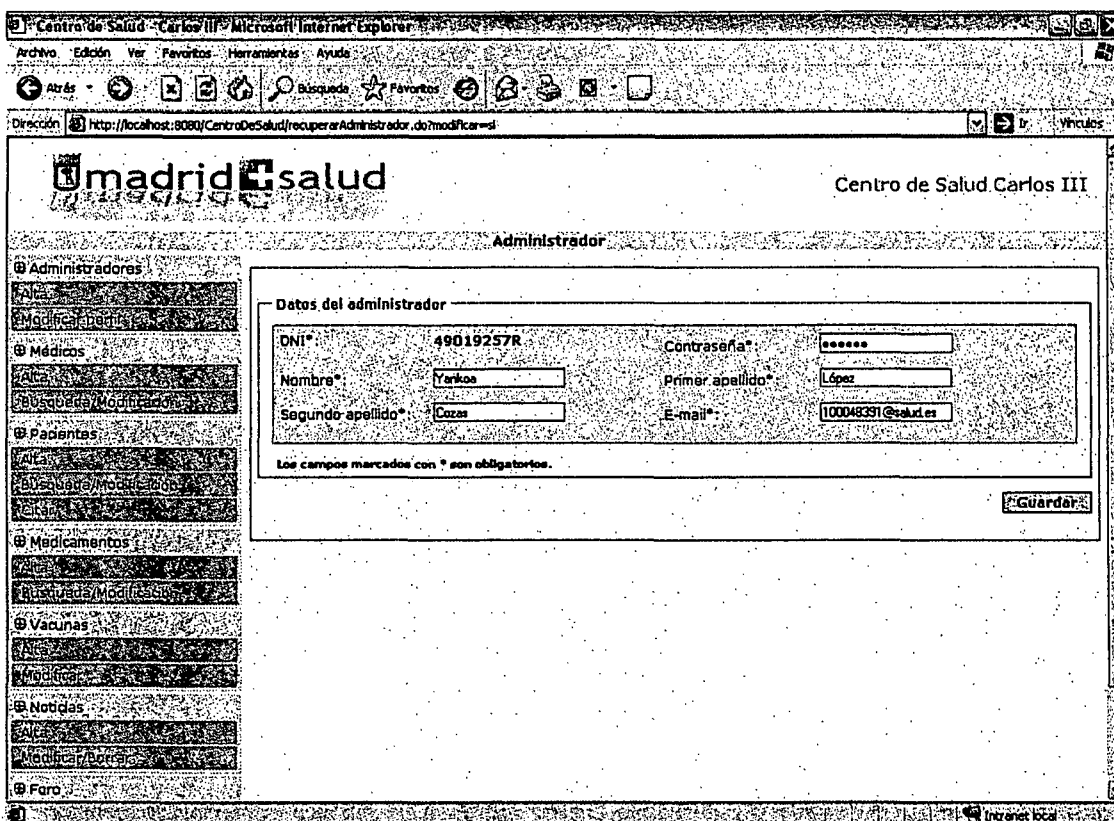
Noticias

Foro

- Figura 148: Administrador guardado con éxito-

5.2.4.2.- MODIFICAR PERFIL

El propio administrador que se haya registrado puede visualizar sus datos personales y modificarlos en caso de ser necesario. Para ello debe pulsar en la opción del menú “Modificar perfil”.



Centro de Salud Carlos III

Administrador

Datos del administrador

ONI*	49019257R	Contraseña*	*****
Nombre*	Yankoa	Primer apellido*	López
Segundo apellido*	Cozas	E-mail*	100048391@salud.es

Los campos marcados con * son obligatorios.

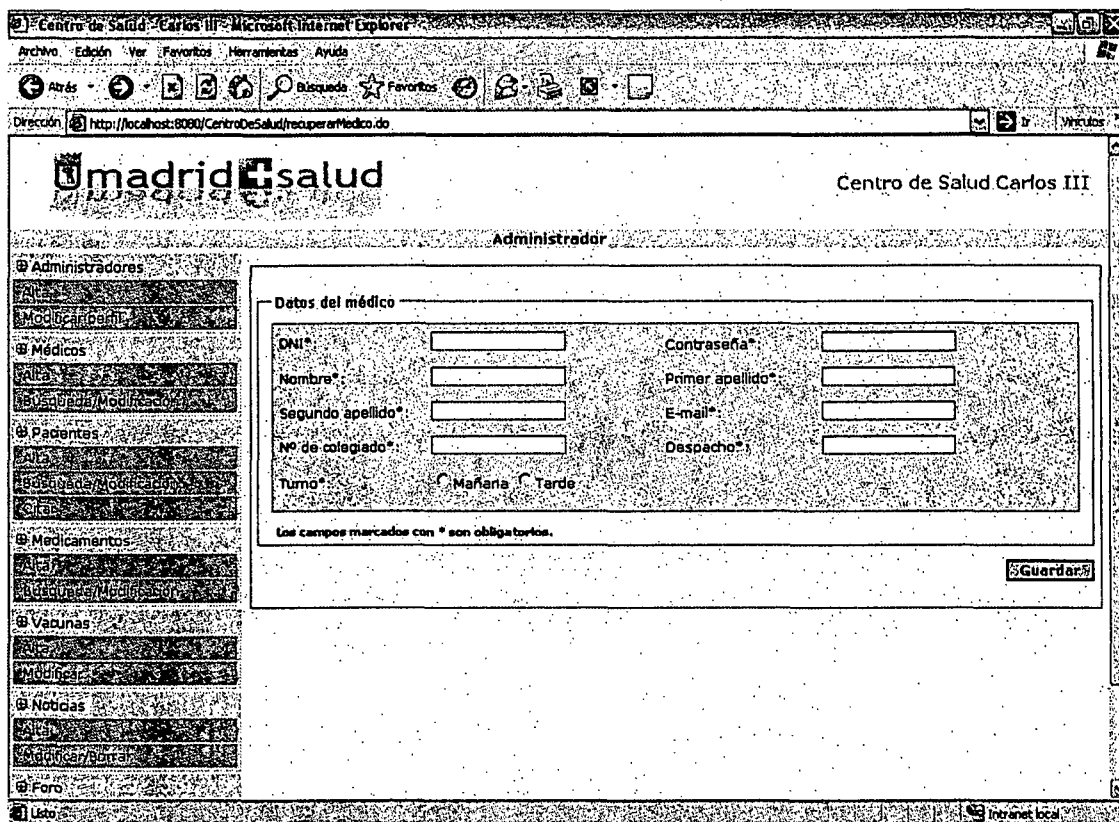
Guardar

- Figura 149: Modificar perfil -

Si desea modificar algún dato, tras hacerlo debe pulsar sobre guardar. En caso de producirse algún error se avisará dando la oportunidad de solucionarlo. Si todo sale correctamente, saldrá un mensaje de éxito.

5.2.4.3.- DAR DE ALTA UN NUEVO MÉDICO

Para dar de alta un nuevo médico, el administrador debe primeramente pulsar en la opción del menú "Alta" (Médicos). De esta forma le aparecerá una pantalla en la que se le pedirá que introduzca los datos personales del médico que va a dar de alta.



Centro de Salud Carlos III

Administrador

Umadridsalud

Datos del médico

DNI* Contraseña*

Nombre* Primer apellido*

Segundo apellido* E-mail*

Nº de colegiado* Despacho*

Turno* ☐ Mañana ☐ Tarde

Los campos marcados con * son obligatorios.

Guardar

Administradores

Alta

Modificar

Médicos

Alta

Modificar

Pacientes

Alta

Modificar

Medicamentos

Alta

Modificar

Vacunas

Alta

Modificar

Noticias

Alta

Modificar

Foro

Alta

Modificar

- Figura 150: Alta médico -

Una vez introduzca los campos necesarios, el administrador pulsará en guardar. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo en los datos introducidos, el administrador podrá modificarlos y tras pulsar de nuevo en guardar estos se actualizarán.



Centro de Salud Carlos III

Administrador

Los datos se han guardado correctamente.

Datos del médico

DNI* 65018567Q Contraseña* *****

Nombre* María Primer apellido* García

Segundo apellido* Pérez E-mail* maria_gg@salud.es

Nº de colegiado* 128 Despacho* 4

Turno* ☒ Mañana ☐ Tarde

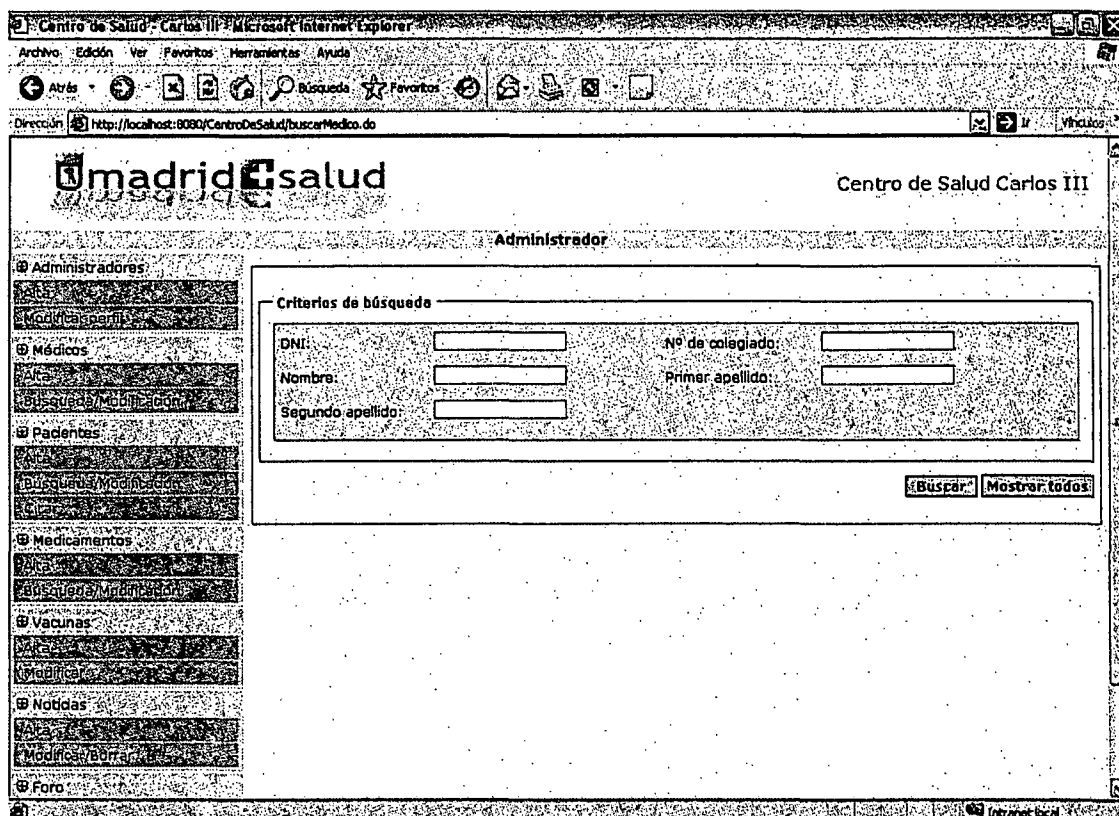
Los campos marcados con * son obligatorios.

Guardar

- Figura 151: Médico guardado con éxito -

5.2.4.4.- CONSULTAR MÉDICO

Cualquier administrador puede consultar todos los médicos que hay en el centro de salud, ya sea para visualizar sus datos y/o actualizarlos. Para ello deberá pulsar en la opción del menú "Búsqueda/Modificación" (Médicos), apareciéndole por tanto la siguiente pantalla.

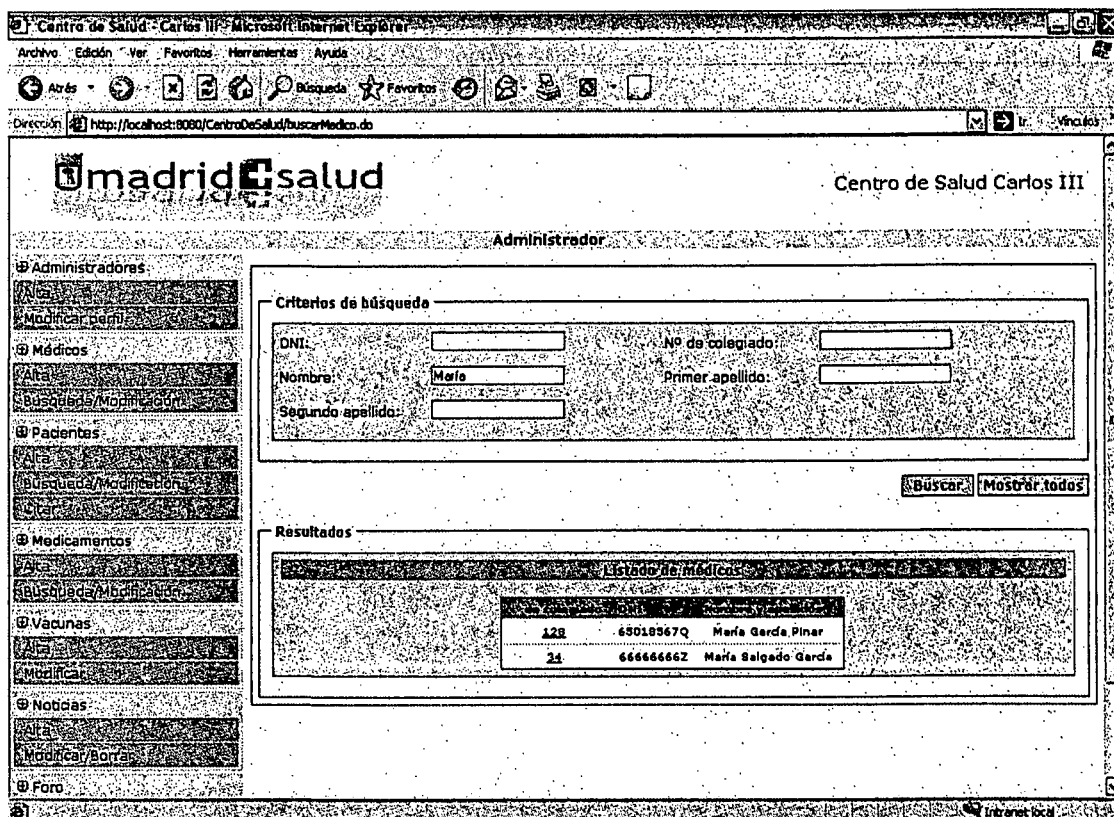


- Figura 152: Consultar médico -

En esta pantalla se presentan dos opciones:

- Buscar a un médico introduciendo algún/os de los criterios de búsqueda (DNI, nombre...) y pulsar buscar, de forma que se muestren todos los médicos que coincidan con los criterios establecidos.
- Pulsar sobre el botón "Mostrar todos" para que aparezcan todos los médicos del centro.

En ambas opciones, si no existen resultados que mostrar, aparecerá un mensaje de aviso.



Centro de Salud Carlos III

Administrador

Administradores

Médicos

Padentes

Medicamentos

Vacunas

Noticias

Foro

Criterios de búsqueda

DNI: Nº de colegiado:

Nombre: Primer apellido:

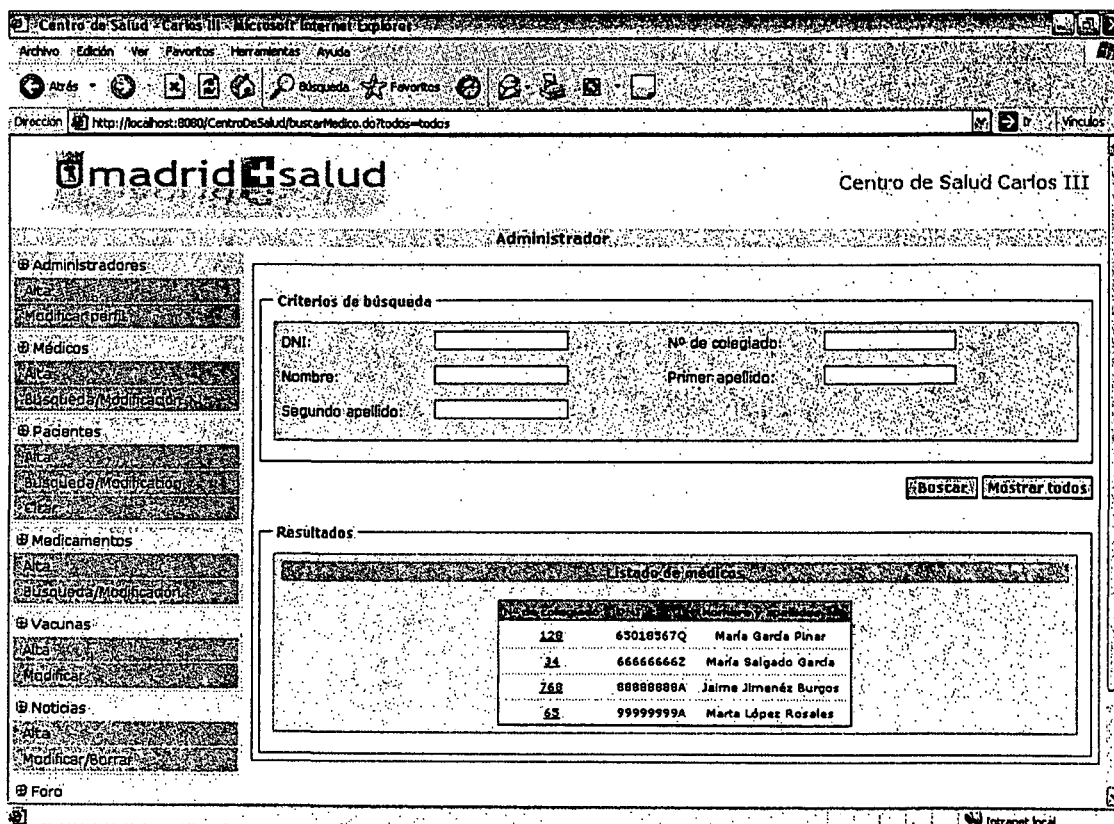
Segundo apellido:

Buscar Mostrar todos

Resultados

Listado de médicos			
128	65018567Q	María García Pinar	
24	66666666Z	María Salgado García	

- Figura 153: Consultar médico introduciendo algún criterio -



Centro de Salud Carlos III

Administrador

Administradores

Médicos

Padentes

Medicamentos

Vacunas

Noticias

Foro

Criterios de búsqueda

DNI: Nº de colegiado:

Nombre: Primer apellido:

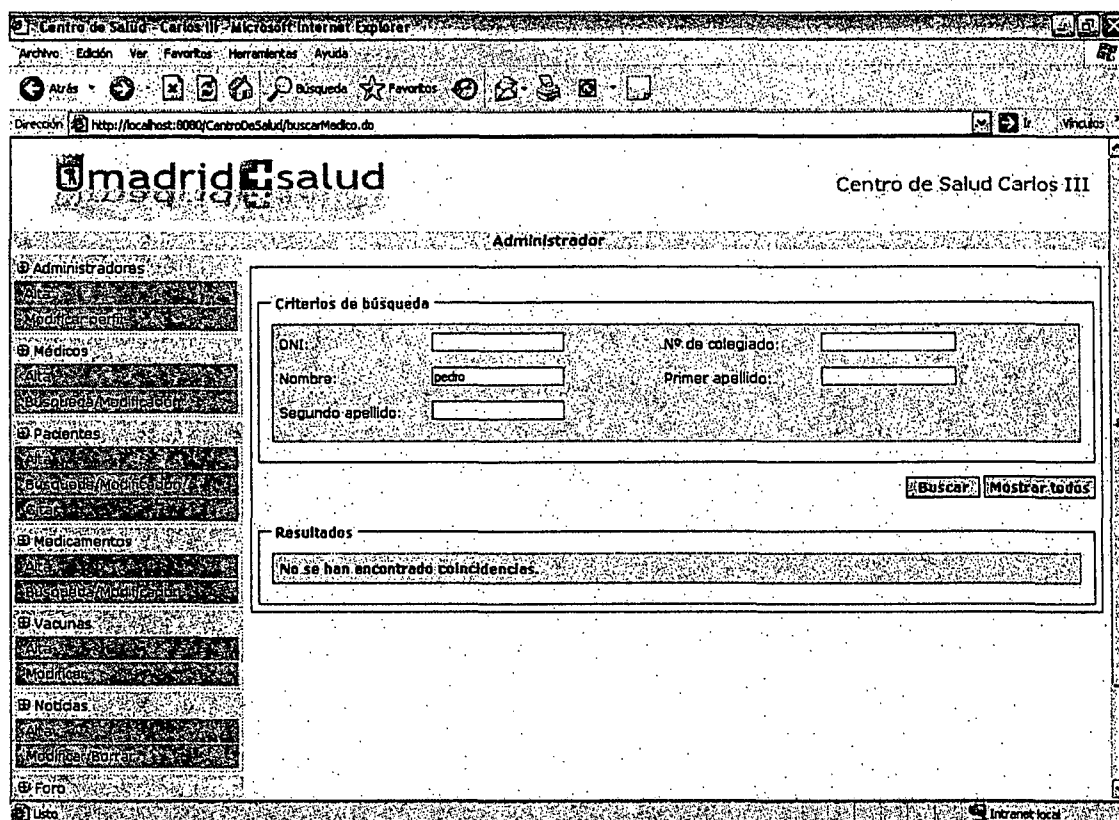
Segundo apellido:

Buscar Mostrar todos

Resultados

Listado de médicos			
128	65018567Q	María García Pinar	
24	66666666Z	María Salgado García	
768	88888888A	Jaime Jiménez Burgos	
63	99999999A	Marta López Rosales	

- Figura 154: Mostrar todos los médicos -



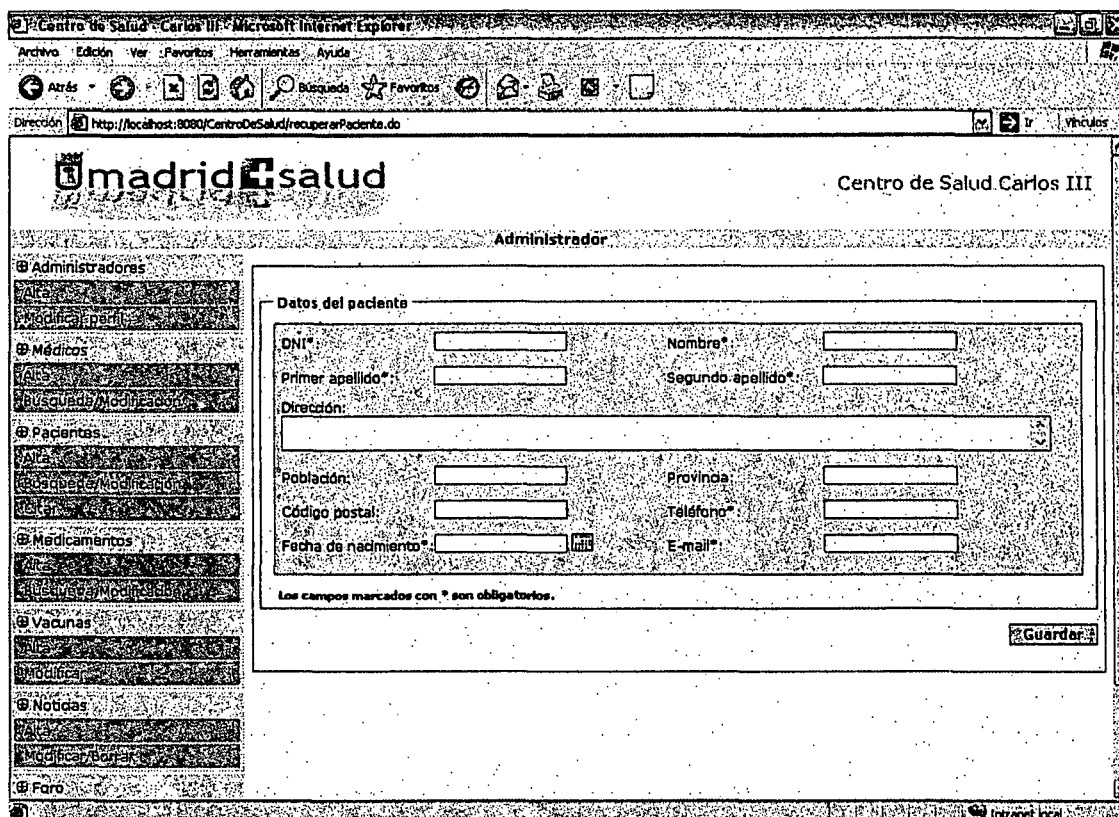
The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/CentroDeSalud/buscarMedico.do'. The page title is 'Centro de Salud Carlos III'. The main content area is titled 'Administrador' and features a search form under the heading 'Criterios de búsqueda'. The form includes input fields for 'DNI:', 'Nº de colegiado:', 'Nombre:' (with the value 'pedro' entered), 'Primer apellido:', and 'Segundo apellido:'. Below the form are two buttons: 'Buscar' and 'Mostrar todos'. The results section, titled 'Resultados', displays the message 'No se han encontrado coincidencias.' (No coincidences found). On the left side of the page, there is a vertical menu with various options: 'Administradores', 'Médicos', 'Pacientes', 'Medicamentos', 'Vacunas', 'Noticias', and 'Foro'. The bottom of the page shows the status bar with 'Internet local'.

- Figura 155: Búsqueda de médico sin coincidencias -

Si tras la búsqueda deseamos ver los datos de un médico, solo debemos pulsar sobre el número de colegiado, apareciendo así una pantalla con sus datos, en la cual el administrador puede modificar alguno y guardarlo de igual manera que en el alta del médico.

5.2.4.5.- DAR DE ALTA UN NUEVO PACIENTE

Para dar de alta un nuevo paciente, el administrador debe primeramente pulsar en la opción del menú "Alta" (Pacientes). De esta forma le aparecerá una pantalla en la que se le pedirá que introduzca los datos personales del paciente que va a dar de alta.



Administrador

Datos del paciente

DNI* Nombre*

Primer apellido* Segundo apellido*

Dirección:

Población: Provincia:

Código postal: Teléfono*

Fecha de nacimiento* E-mail*

Los campos marcados con * son obligatorios.

Guardar

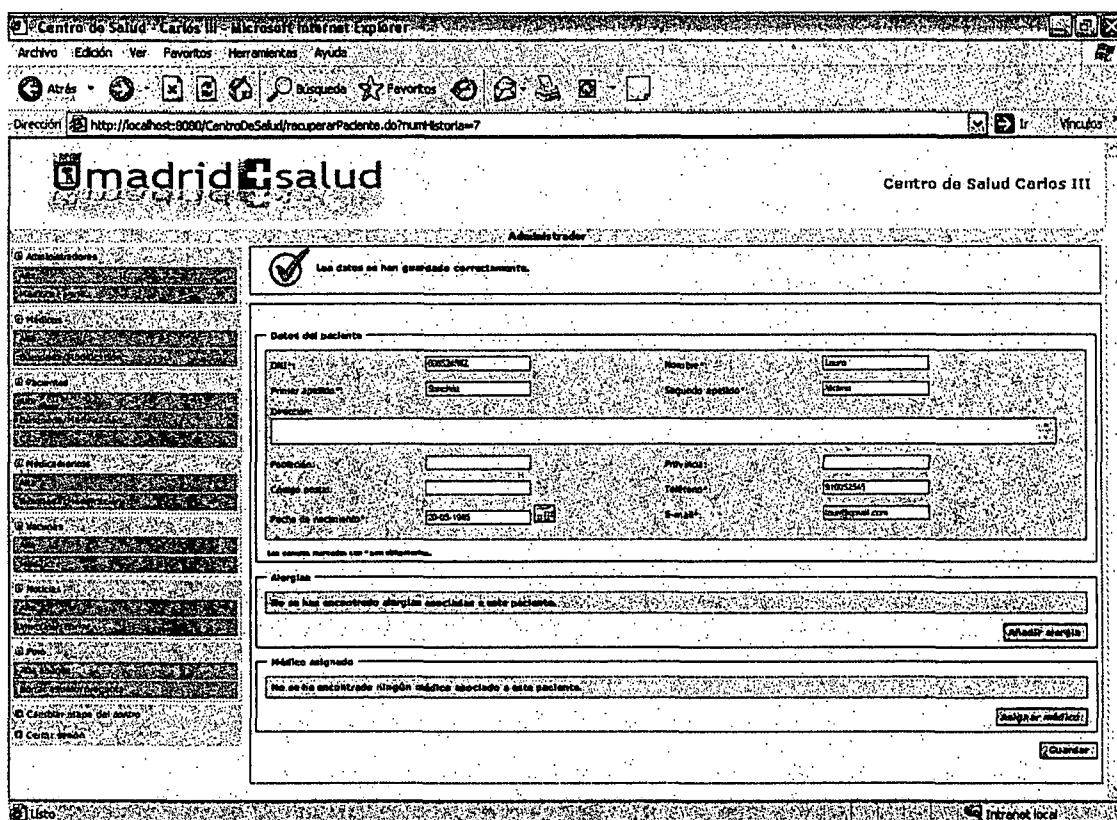
Sidebar:

- Administradores
 - Alta
 - Modificar/eliminar
- Médicos
 - Alta
 - Modificar/eliminar
- Pacientes
 - Alta
 - Modificar/eliminar
- Medicamentos
 - Alta
 - Modificar/eliminar
- Vacunas
 - Alta
 - Modificar/eliminar
- Noticias
 - Alta
 - Modificar/eliminar
- Foro

- Figura 156: Alta paciente -

Una vez introduzca los campos necesarios, el administrador pulsará en guardar. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos, además aparecerán dos nuevas opciones: introducir alergias y asignar médico.

En caso de encontrar algún fallo en los datos introducidos, el administrador podrá modificarlos y tras pulsar de nuevo en guardar estos se actualizarán.



Centro de Salud Carlos III - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Abre Busqueda Favoritos

Dirección http://localhost:8080/CentroDeSalud/recuperarPaciente.do?numHistoria=7

Madrid Salud Centro de Salud Carlos III

Administrador

Los datos se han guardado correctamente.

Datos del paciente

DNI: Nombre: Apellido:

Primer apellido: Segundo apellido:

Dirección:

Provincia: Teléfono:

Código postal: Teléfono:

Fecha de nacimiento: Email:

Los campos marcados con * son obligatorios.

Alergias

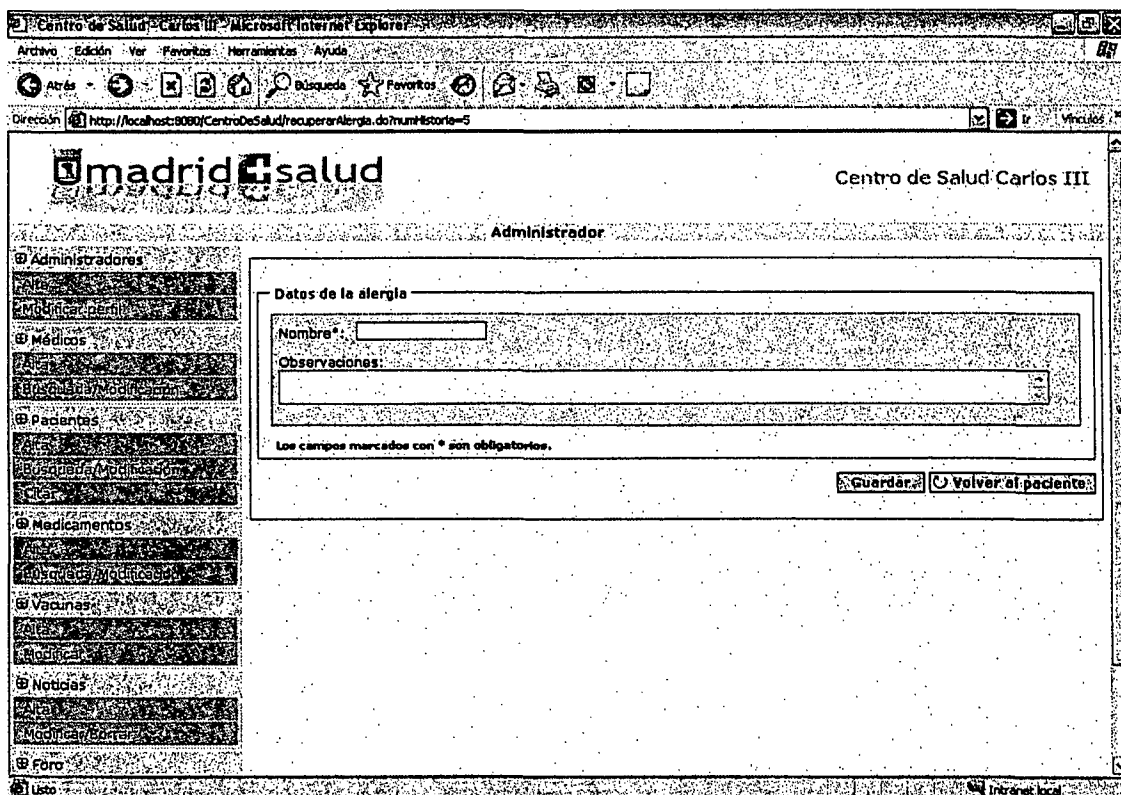
No se ha encontrado alergias asociadas a este paciente.

Médico asignado

No se ha encontrado ningún médico asociado a este paciente.

- Figura 157: Paciente guardado con éxito -

El administrador opcionalmente puede añadir alergias al paciente. Para ello deberá pulsar en el botón “Añadir alergia”, de forma que le aparecerá la siguiente pantalla.



- Figura 158: Añadir alergia -

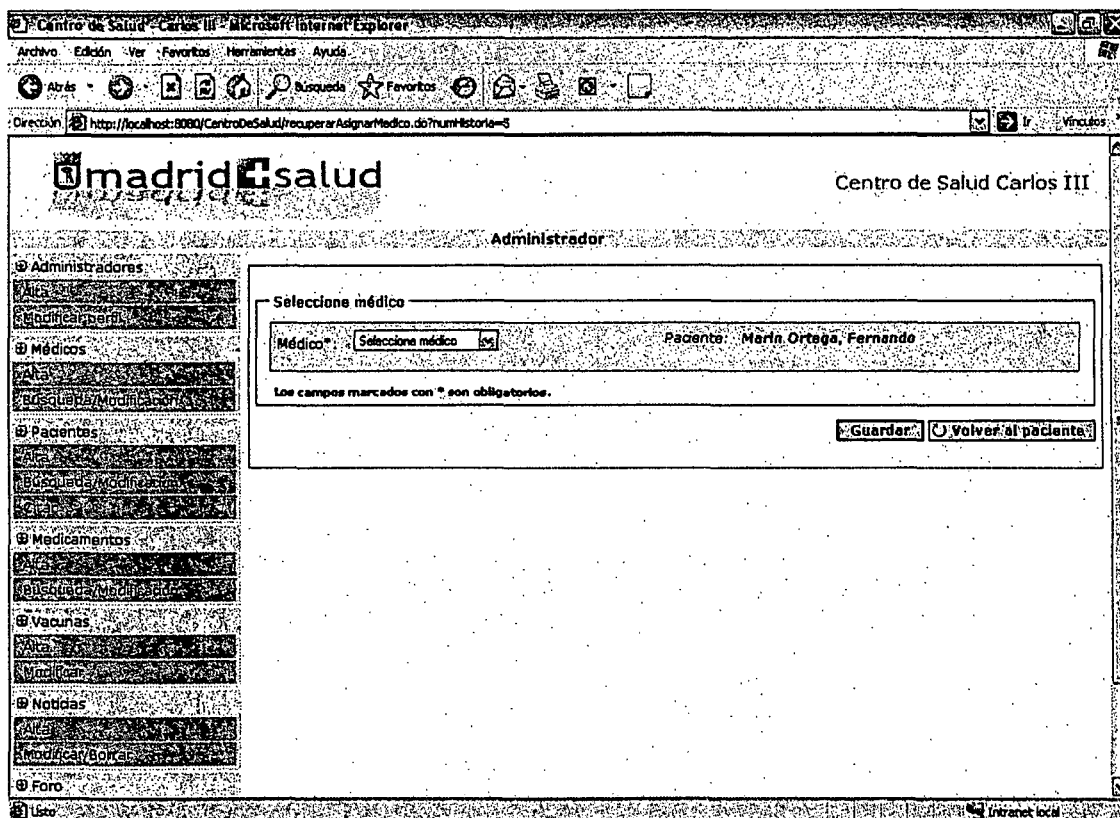
En esta pantalla el administrador podrá introducir los datos necesarios y pulsar sobre guardar para almacenar la alergia, de forma que si se produce algún error se avisará para que pueda solventarlo.

Si no desea introducir ninguna alergia, podrá pulsar en el botón “Volver al paciente” para regresar a la pantalla anterior.

Si el paciente contiene alguna alergia introducida, esta aparecerá en la pantalla junto con sus datos personales, de forma que si el administrador pulsa sobre el nombre de la alergia pueda modificarla, cambiando los datos que sean necesarios y pulsando sobre el botón “Guardar”.

- Figura 159: Datos paciente con alergia -

Página 207 de 315



- Figura 160: Asignar médico -

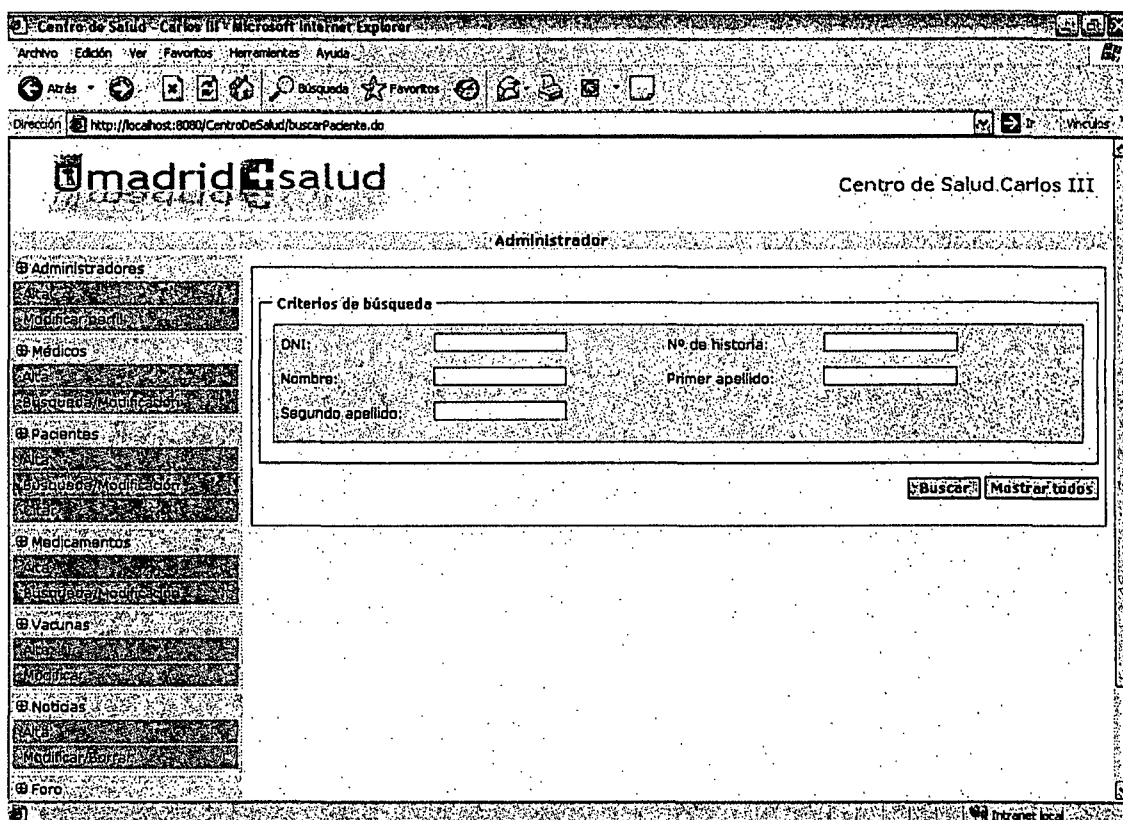
En esta pantalla el administrador podrá introducir los datos necesarios y pulsar sobre guardar para asignar el médico al paciente, de forma que si se produce algún error se avisará para que pueda solventarlo.

Si no desea asignar ningún médico, podrá pulsar en el botón “Volver al paciente” para regresar a la pantalla anterior.

Cuando un paciente tiene un médico asignado, el administrador tendrá opción de reasignarle al paciente otro médico pulsando sobre el botón “Cambiar médico”, de forma que asigne otro médico al paciente y pulse sobre “Guardar”.

- Figura 161: Datos paciente con alergia y médico asignado -

Cualquier administrador puede consultar los pacientes que hay en el centro de salud, ya sea para visualizar sus datos y/o actualizarlos. Para ello deberá pulsar en la opción del menú “Búsqueda/Modificación” (Pacientes), apareciéndole por tanto la siguiente pantalla.

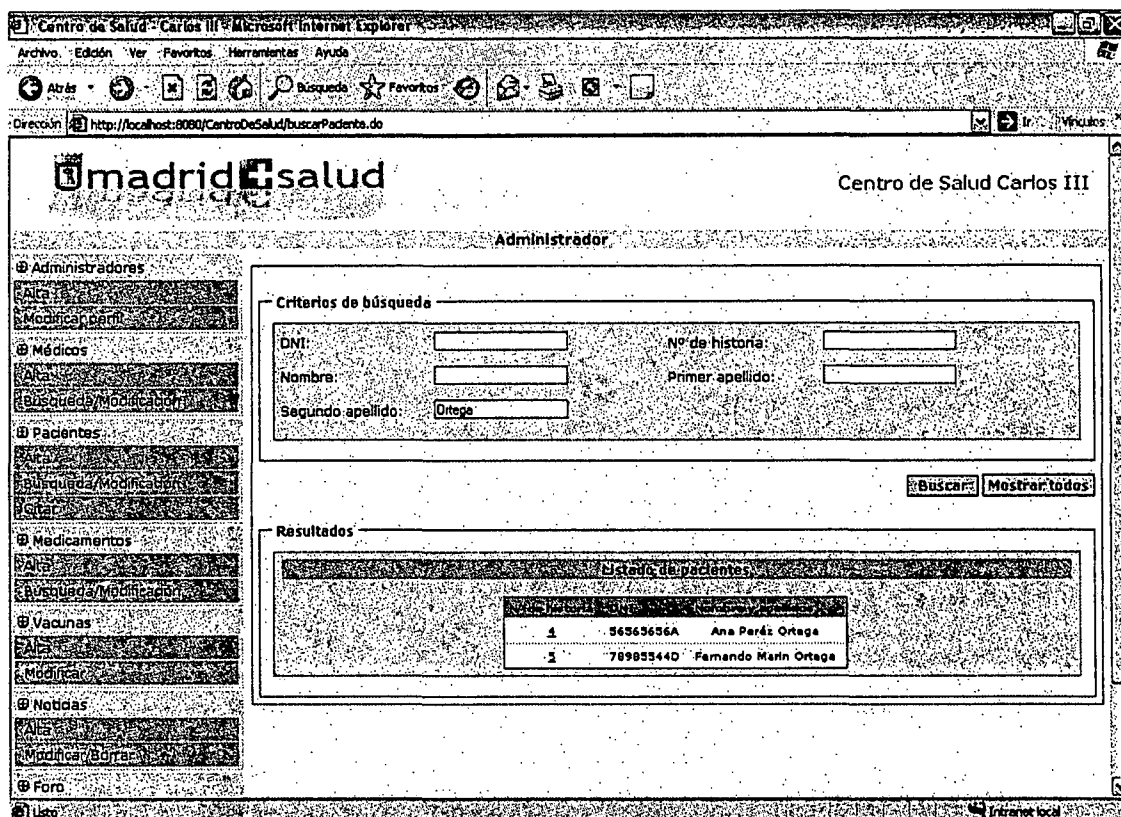


- Figura 162: Consultar paciente -

En esta pantalla se presentan dos opciones:

- Buscar a un paciente introduciendo algún/os de los criterios de búsqueda (DNI, nombre...) y pulsar buscar, de forma que se muestren todos los pacientes que coincidan con los criterios establecidos.
- Pulsar sobre el botón "Mostrar todos" para que aparezcan todos los pacientes del centro.

En ambas opciones, si no existen resultados que mostrar, aparecerá un mensaje de aviso.



Administrador

Criterios de búsqueda

DNI: N° de historia:

Nombre: Primer apellido:

Segundo apellido:

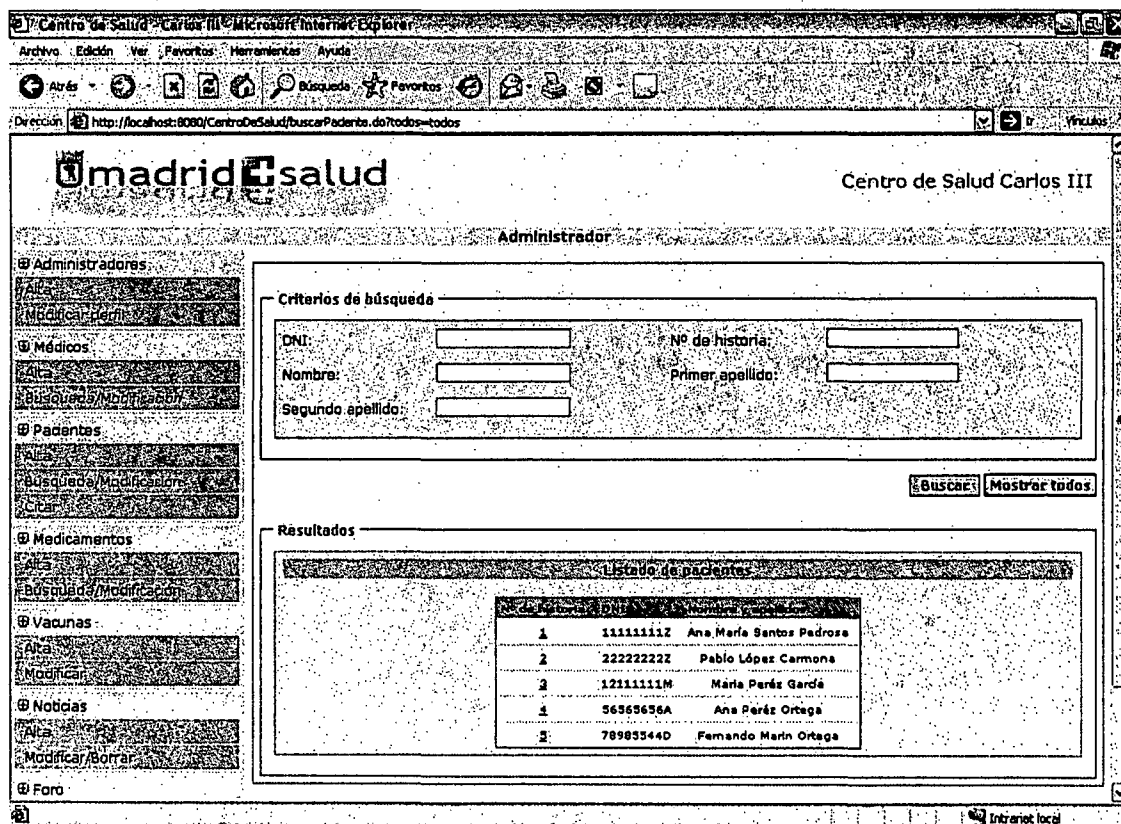
Buscar **Mostrar todos**

Resultados

Listado de pacientes

1	56565656A	Ana Peréz Ortega
2	78985544D	Fernando Marín Ortega

- Figura 163: Consultar paciente introduciendo algún criterio -



Administrador

Criterios de búsqueda

DNI: N° de historia:

Nombre: Primer apellido:

Segundo apellido:

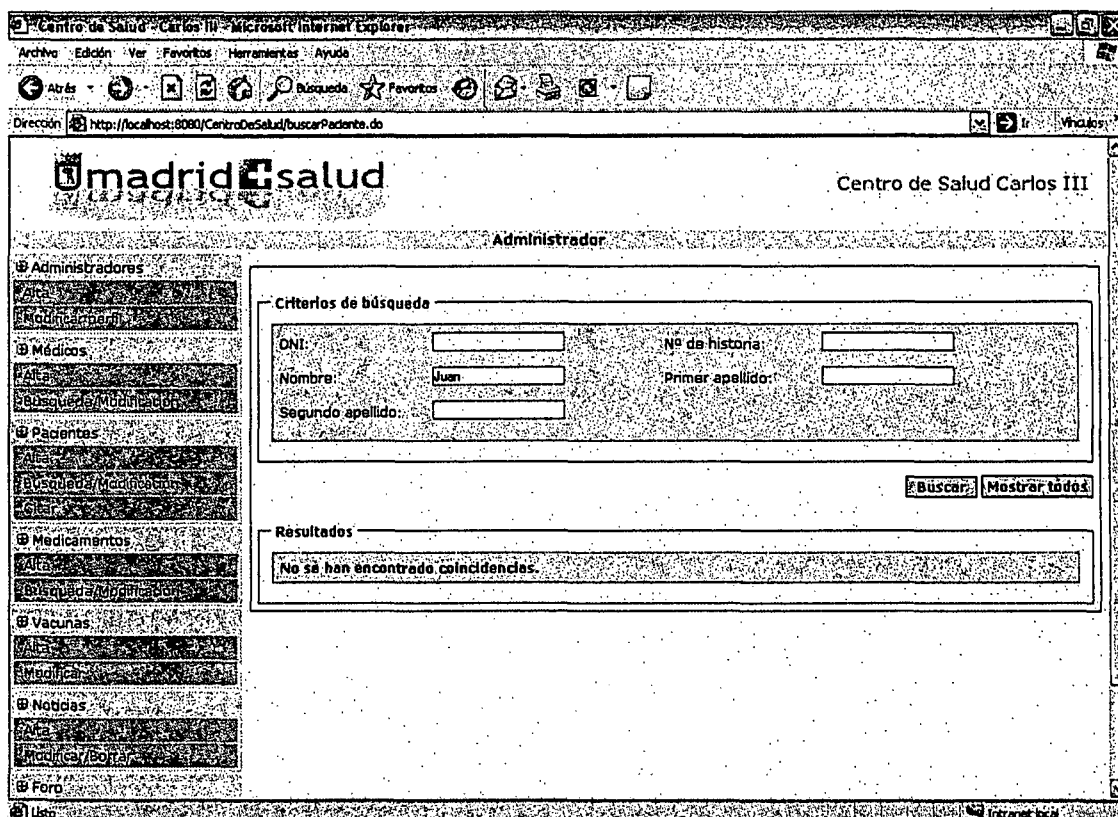
Buscar **Mostrar todos**

Resultados

Listado de pacientes

1	11111111Z	Ana María Santos Padrosa
2	22222222Z	Pablo López Camona
3	12111111M	María Peréz García
4	56565656A	Ana Peréz Ortega
5	78985544D	Fernando Marín Ortega

- Figura 164: Mostrar todos los pacientes -



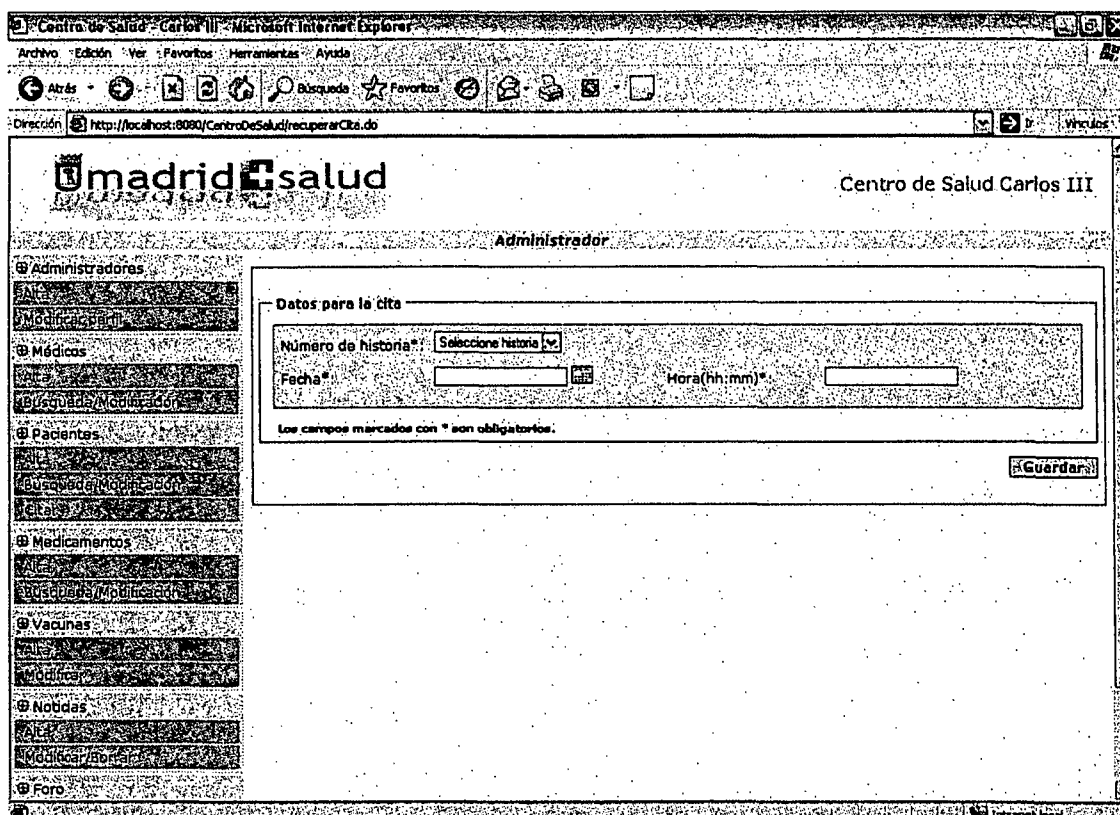
The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/CentroDeSalud/buscarPaciente.do'. The page title is 'Centro de Salud Carlos III'. The main content area is titled 'Administrador' and features a search form. The search form has a section 'Criterios de búsqueda' with fields for 'ONI', 'Nº de historia', 'Nombre' (containing 'Juan'), 'Primer apellido', and 'Segundo apellido'. Below the search form are buttons for 'Buscar' and 'Mostrar todos'. The results section, titled 'Resultados', displays the message 'No se han encontrado coincidencias.'.

- Figura 165: Búsqueda de paciente sin coincidencia -

Si tras la búsqueda deseamos ver los datos de un paciente, solo debemos pulsar sobre el número de historia del paciente que deseamos consultar, apareciendo así una pantalla con sus datos, en la cual el administrador podrá modificar alguno y guardarlo de igual manera que en el alta del paciente, además de poder añadir nuevas alergias y/o asignar/cambiar médico.

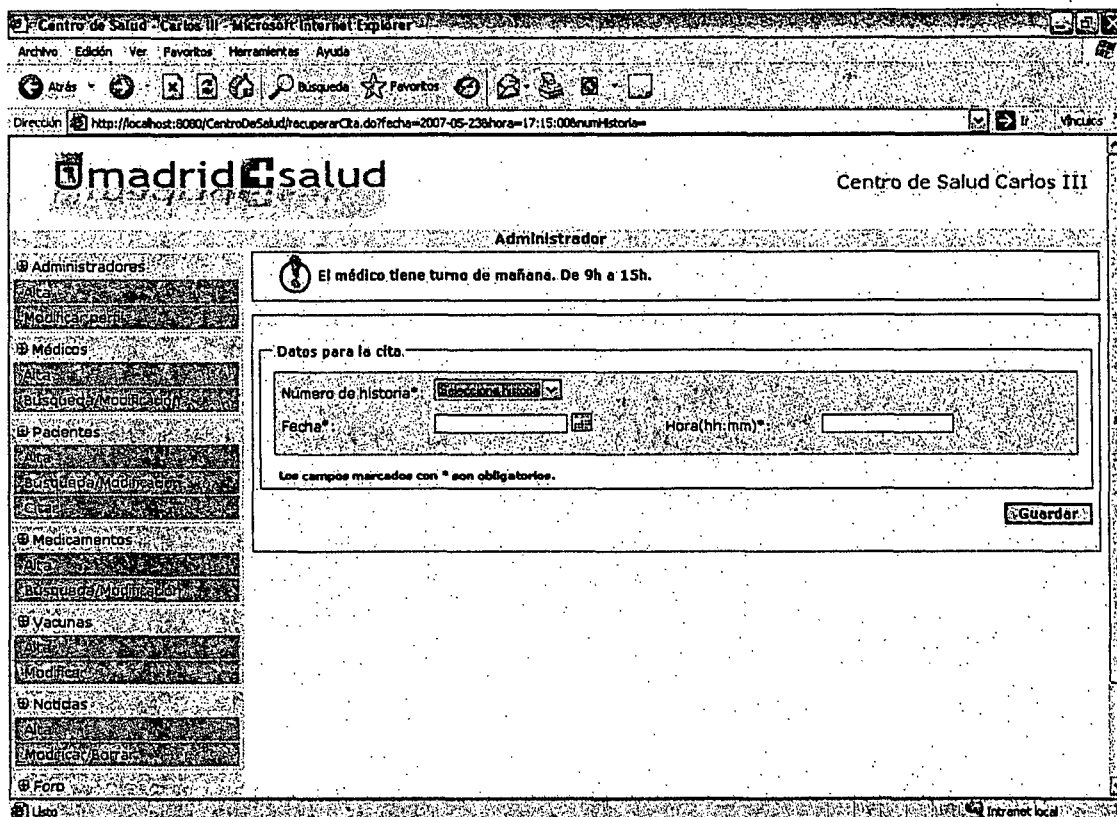
5.2.4.7.- CITAR PACIENTE

Cualquier administrador podrá citar a un paciente (generalmente dar la primera cita), pulsando sobre la opción del menú "Citar paciente", apareciendo la siguiente pantalla.



- Figura 166: Administrador cita a paciente -

Tras introducir seleccionar al paciente que desea citar e introducir la fecha y hora de la cita, debe pulsar “Guardar” para generar el volante. Si se produce algún error, saldrá un aviso permitiendo así al administrador solucionar dicho error. Si todo sale correctamente, aparecerá el volante de cita, de forma que si el administrador lo desea pueda pulsar sobre el botón “Imprimir” para dar el volante al paciente.



Centro de Salud Carlos III

Administrador

El médico tiene turno de mañana. De 9h a 15h.

Datos para la cita:

Número de historia*

Fecha* Hora(hh:mm)*

Los campos marcados con * son obligatorios.

Administradores

Médicos

Pacientes

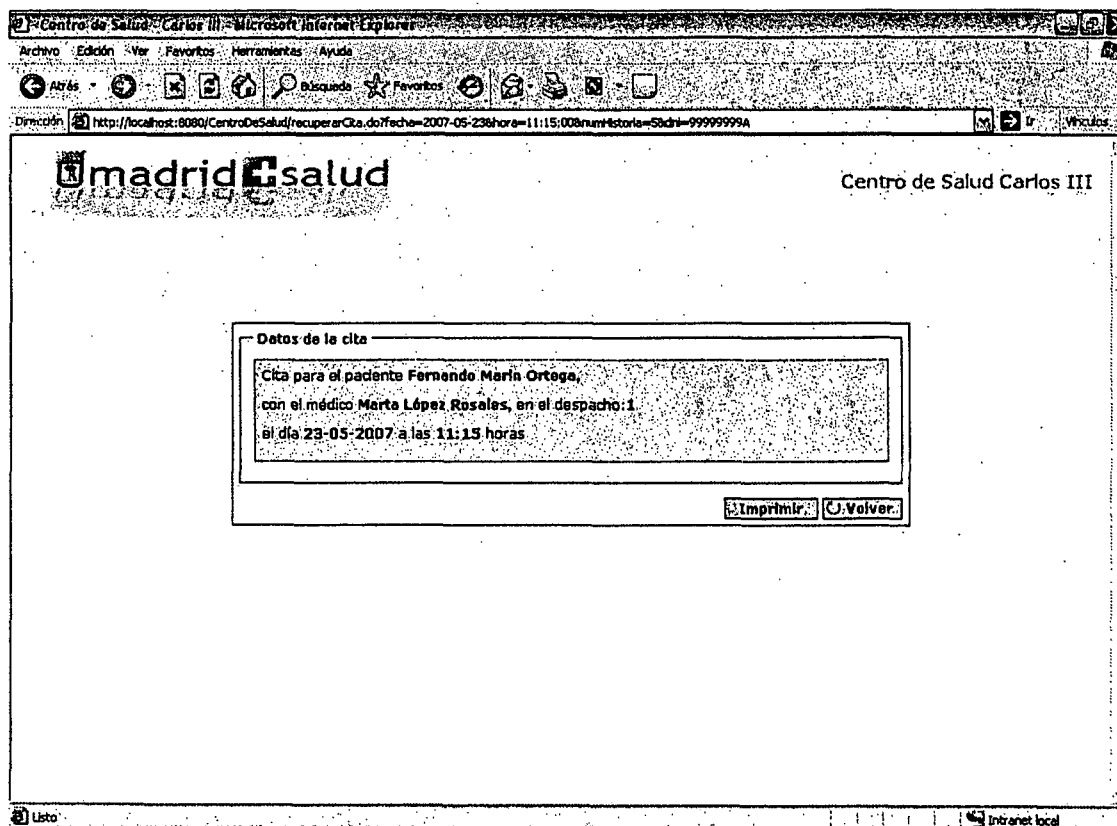
Medicamentos

Vacunas

Noticias

Foro

- Figura 167: Error al citar al paciente -



Centro de Salud Carlos III

Datos de la cita

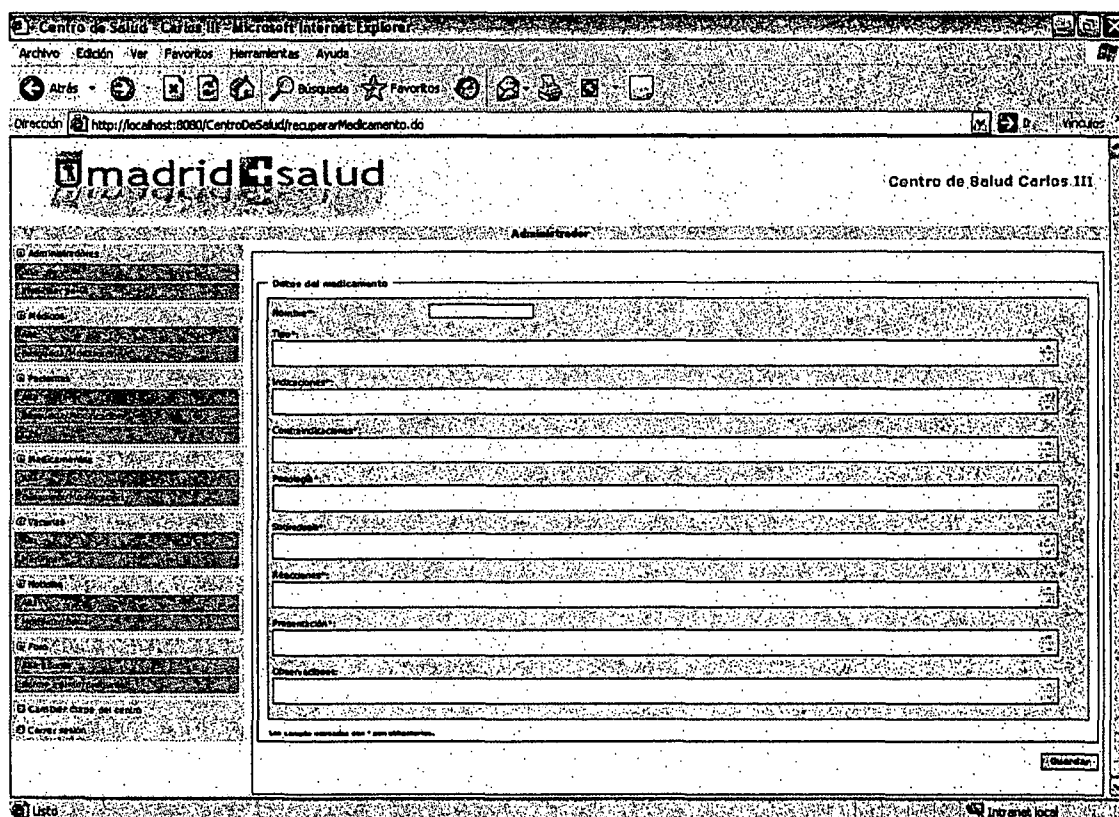
Cita para el paciente Fernando Marín Ortega,
con el médico Marta López Rosales, en el despacho:1
el día 23-05-2007 a las 11:15 horas

- Figura 168: Volante de cita -

Tras finalizar el proceso, el administrador pulsará el botón “Volver” que le permitirá dar más cita o seguir navegando por el menú.

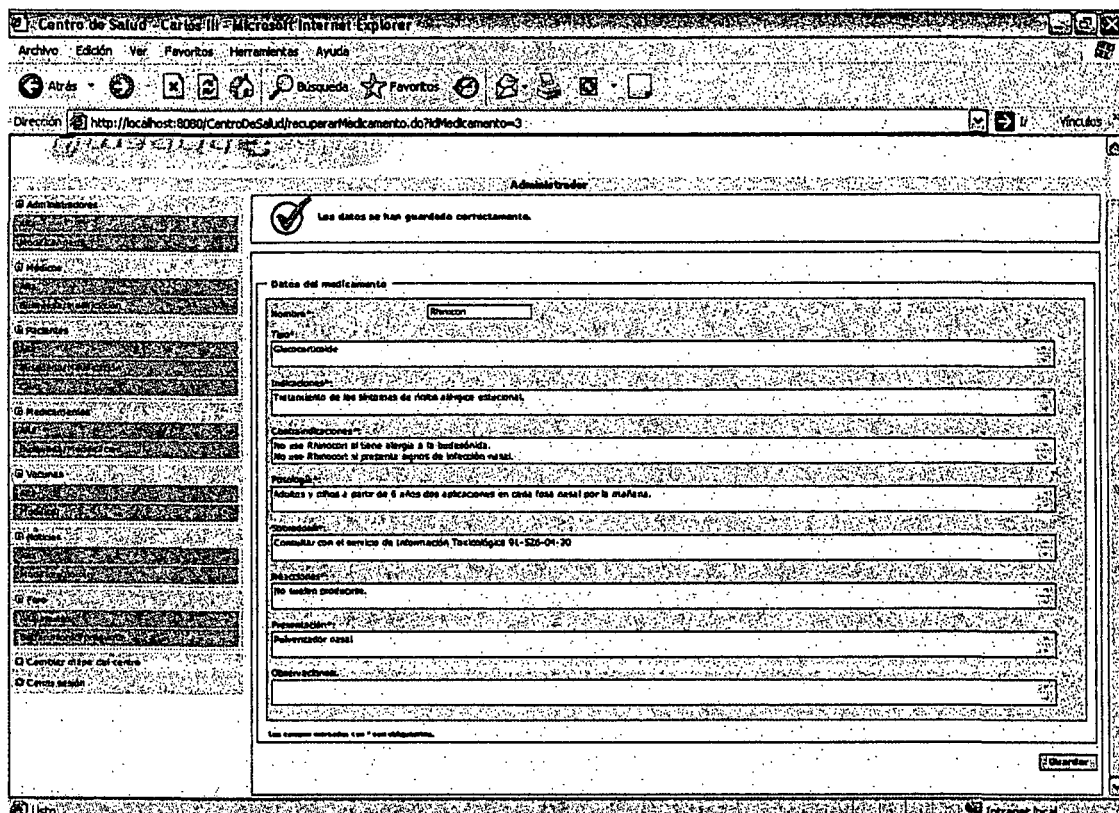
5.2.4.8.- DAR DE ALTA UN NUEVO MEDICAMENTO

Para dar de alta un nuevo medicamento, el administrador debe primeramente pulsar en la opción del menú “Alta” (Medicamentos). De esta forma le aparecerá una pantalla en la que se le pedirá que introduzca los datos del medicamento que va a dar de alta.



- Figura 169: Alta medicamento -

Una vez introduzca los campos necesarios, el administrador pulsará en guardar. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo en los datos introducidos, el administrador podrá modificarlos y tras pulsar de nuevo en guardar estos se actualizarán.

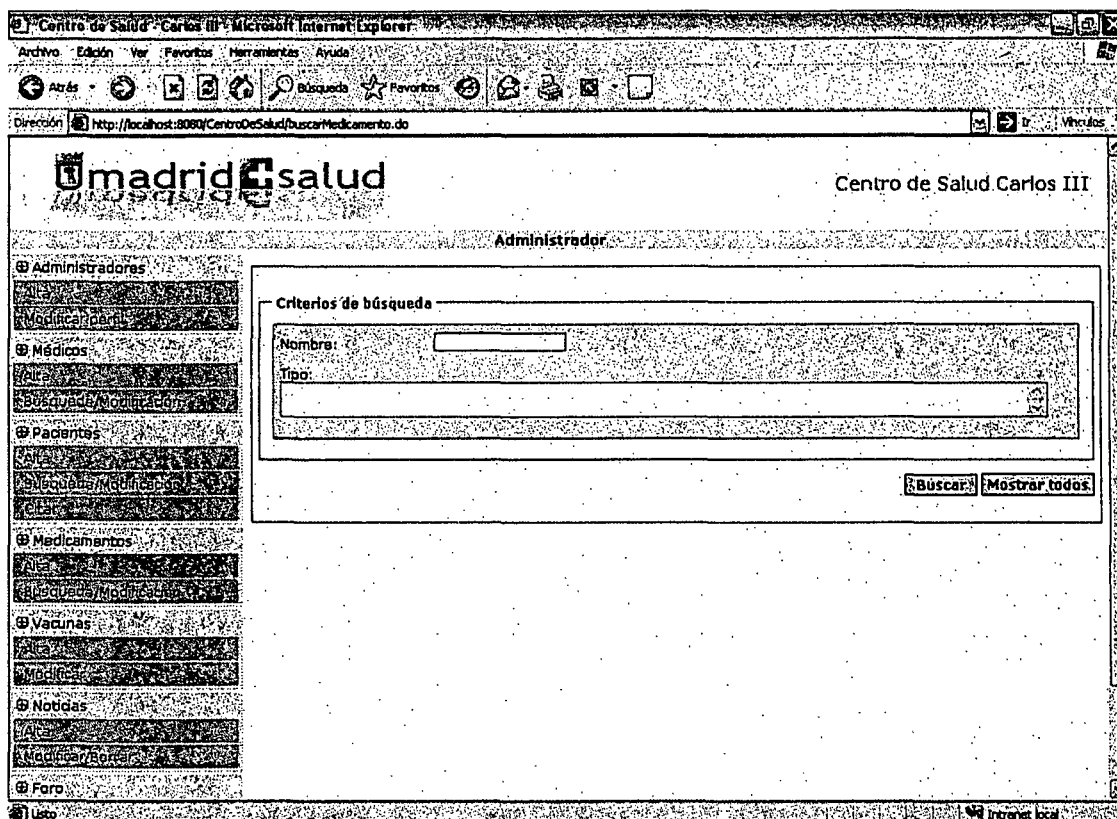


The screenshot shows a web browser window titled "Centro de Salud - Carlos III - Microsoft Internet Explorer". The address bar shows the URL: <http://localhost:8080/CentroDeSalud/recuperarMedicamento.do?idMedicamento=3>. The page displays a confirmation message: "Los datos se han guardado correctamente." (The data has been saved correctly). Below this, there is a form titled "Datos del medicamento" (Medication data) with various input fields and checkboxes. The form includes sections for "Indicaciones" (Indications), "Contraindicaciones" (Contraindications), "Precauciones" (Precautions), and "Efectos secundarios" (Side effects). The form is currently empty, and a "Guardar" (Save) button is visible at the bottom right.

- Figura 170: Medicamento guardado con éxito -

5.2.4.9.- CONSULTAR MEDICAMENTO

Cualquier administrador puede consultar los medicamentos, ya sea para visualizar sus datos y/o actualizarlos. Para ello deberá pulsar en la opción del menú "Búsqueda/Modificación" (Medicamentos), apareciéndole por tanto la siguiente pantalla.

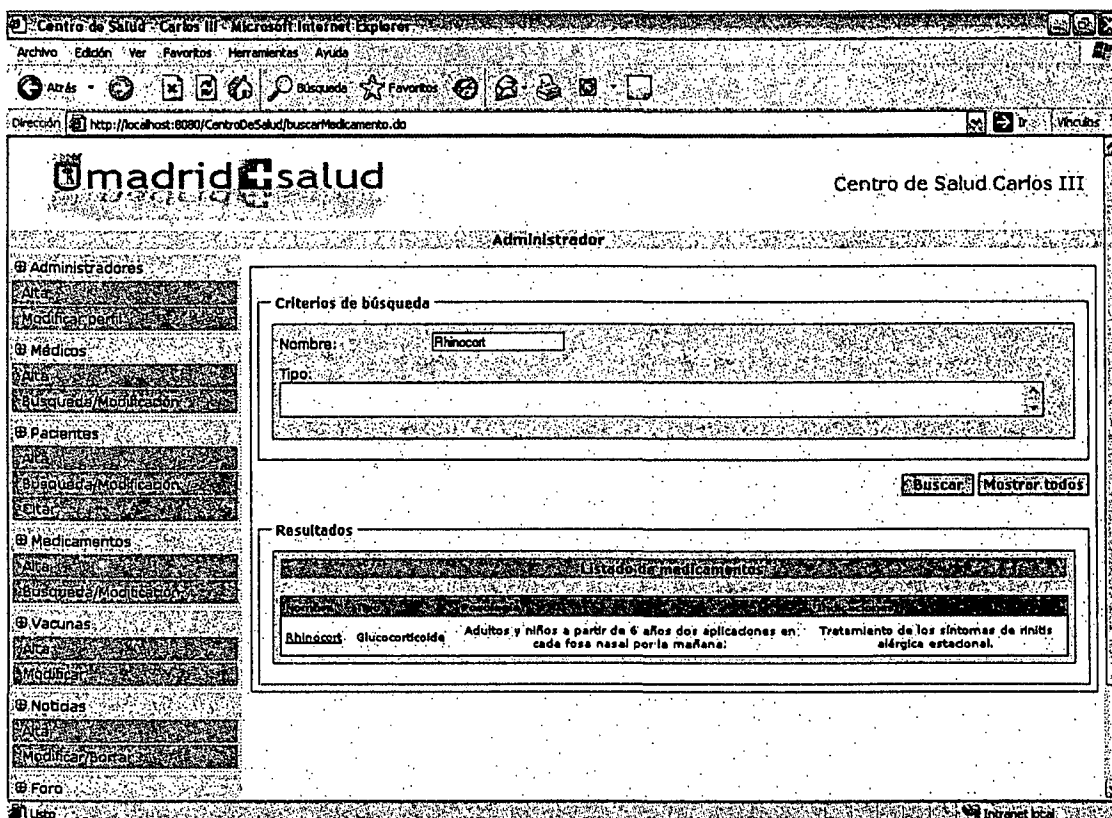


- Figura 171: Consultar medicamento -

En esta pantalla se presentan dos opciones:

- Buscar un medicamento introduciendo algún/os de los criterios de búsqueda (nombre, tipo) y pulsar buscar, de forma que se muestren todos los medicamentos que coincidan con los criterios establecidos.
- Pulsar sobre el botón “Mostrar todos” para que aparezcan todos los medicamentos de la base de datos.

En ambas opciones, si no existen resultados que mostrar, aparecerá un mensaje de aviso.



Centro de Salud Carlos III - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atás Búsqueda Favoritos

Dirección http://localhost:8080/CentroDeSalud/buscarMedicamento.do

Madrid Salud Centro de Salud Carlos III

Administrador

Criterios de búsqueda

Nombre:

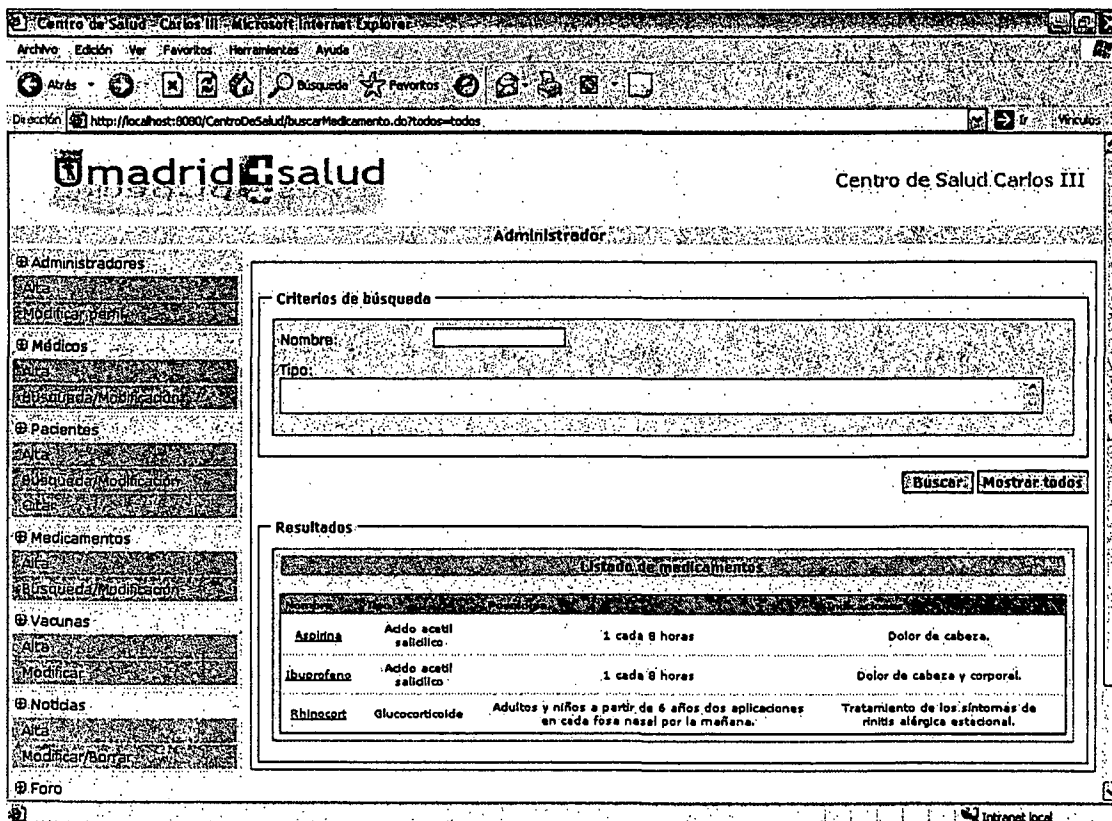
Tipo:

Resultados

Estado de medicamentos			
Rhincort	Glucocorticoide	Adultos y niños a partir de 6 años dos aplicaciones en cada fosa nasal por la mañana.	Tratamiento de los síntomas de rinitis alérgica estacional.

Usted

- Figura 172: Consultar medicamento introduciendo algún criterio -



Centro de Salud Carlos III - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atás Búsqueda Favoritos

Dirección http://localhost:8080/CentroDeSalud/buscarMedicamento.do?todos=todos

Madrid Salud Centro de Salud Carlos III

Administrador

Criterios de búsqueda

Nombre:

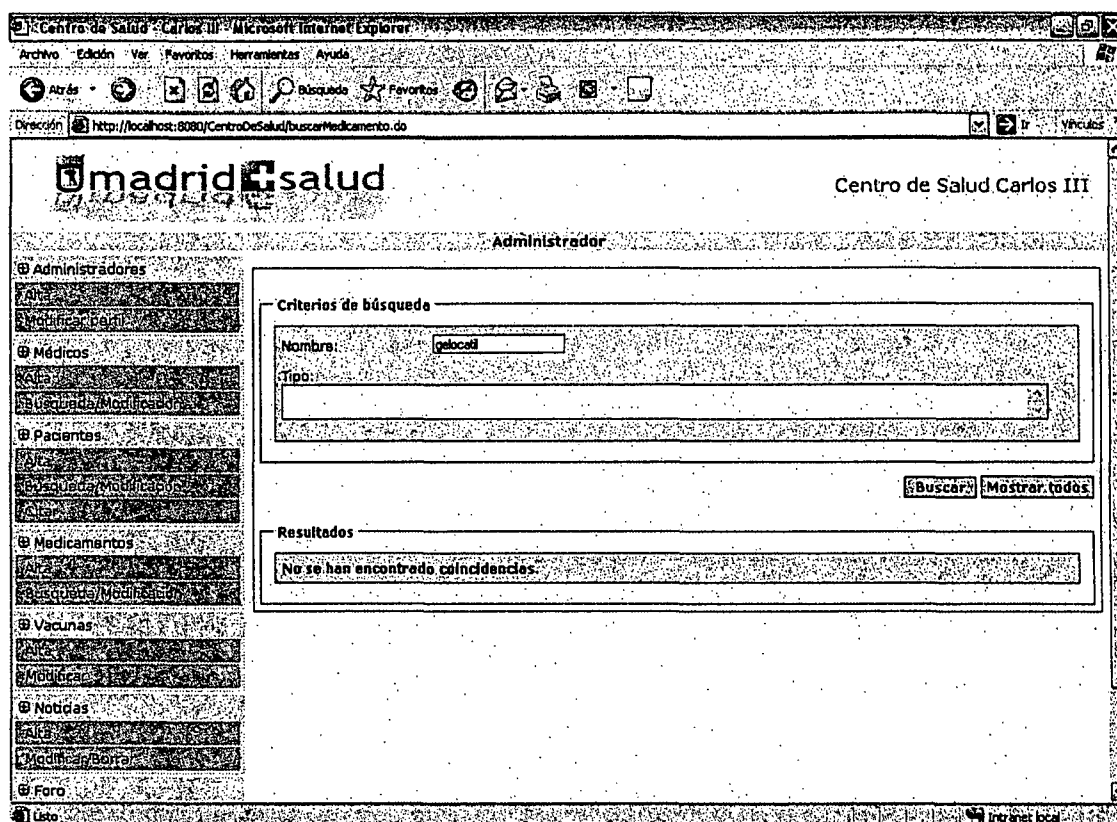
Tipo:

Resultados

Estado de medicamentos			
Aspirina	Acido acetil salicílico	1 cada 8 horas	Dolor de cabeza.
Ibuprofeno	Acido acetil salicílico	1 cada 8 horas	Dolor de cabeza y corporal.
Rhincort	Glucocorticoide	Adultos y niños a partir de 6 años dos aplicaciones en cada fosa nasal por la mañana.	Tratamiento de los síntomas de rinitis alérgica estacional.

Usted

- Figura 173: Mostrar todos los medicamentos -

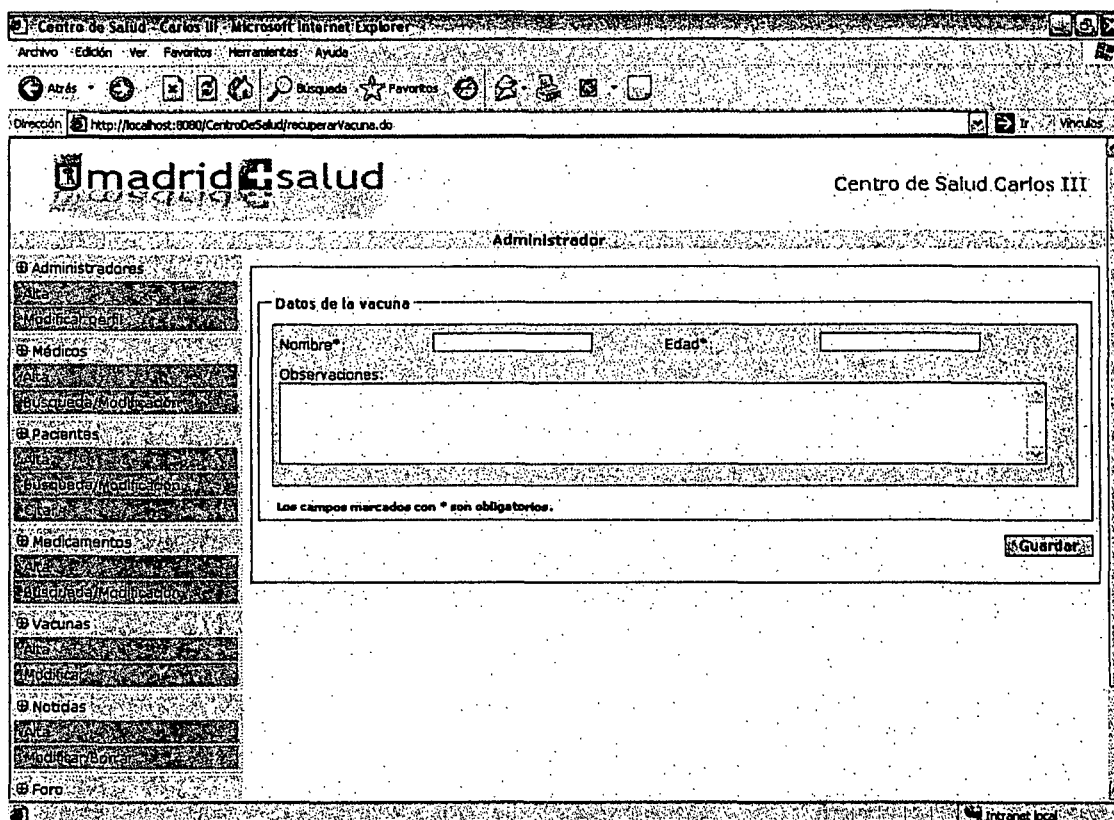


- Figura 174: Búsqueda de medicamento sin coincidencias -

Si tras la búsqueda deseamos ver los datos de un medicamento, solo debemos pulsar sobre su nombre, apareciendo así una pantalla con sus datos, en la cual el administrador podrá modificar alguno y guardarlo de igual manera que en el alta del medicamento.

5.2.4.10.- DAR DE ALTA UNA NUEVA VACUNA

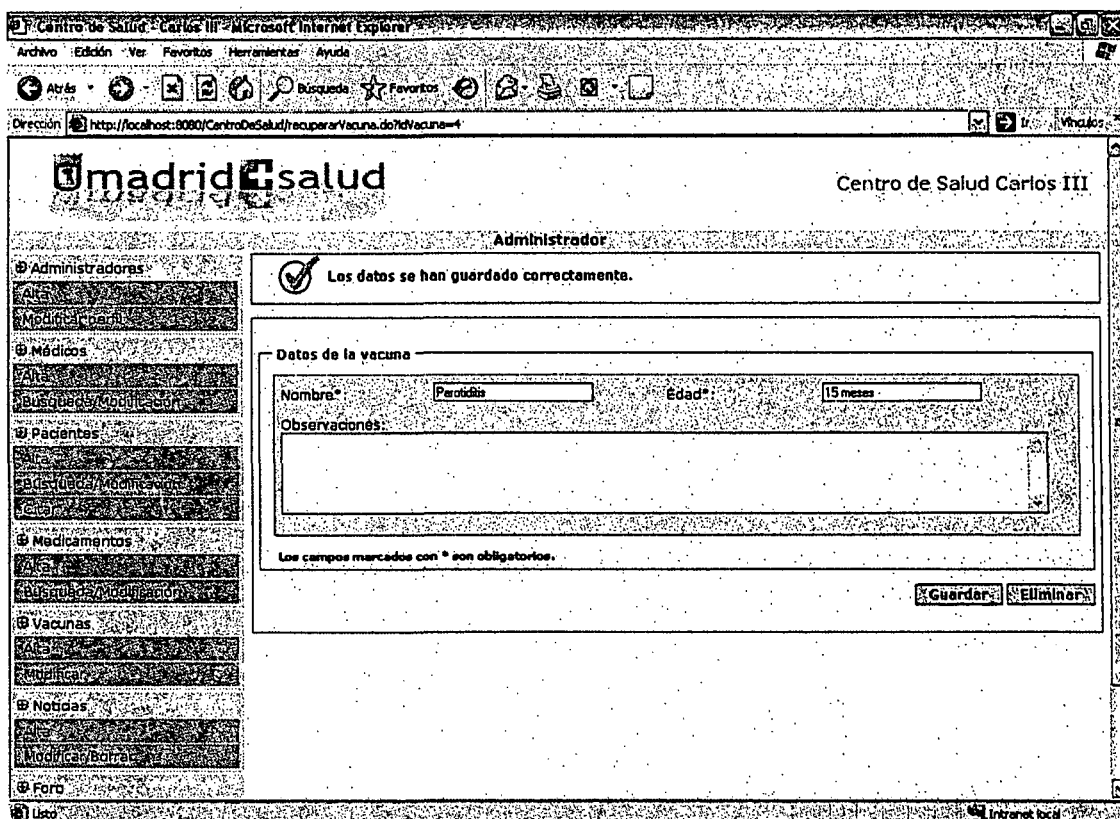
Para dar de alta una nueva vacuna, el administrador debe primeramente pulsar en la opción del menú "Alta" (Vacunas). De esta forma le aparecerá una pantalla en la que se le pedirá que introduzca los datos de la vacuna que va a dar de alta.



- Figura 175: Alta vacuna -

Una vez introduzca los campos necesarios, el administrador pulsará sobre “Guardar”. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo en los datos introducidos, el administrador podrá modificarlos y tras pulsar de nuevo en “Guardar” estos se actualizarán.

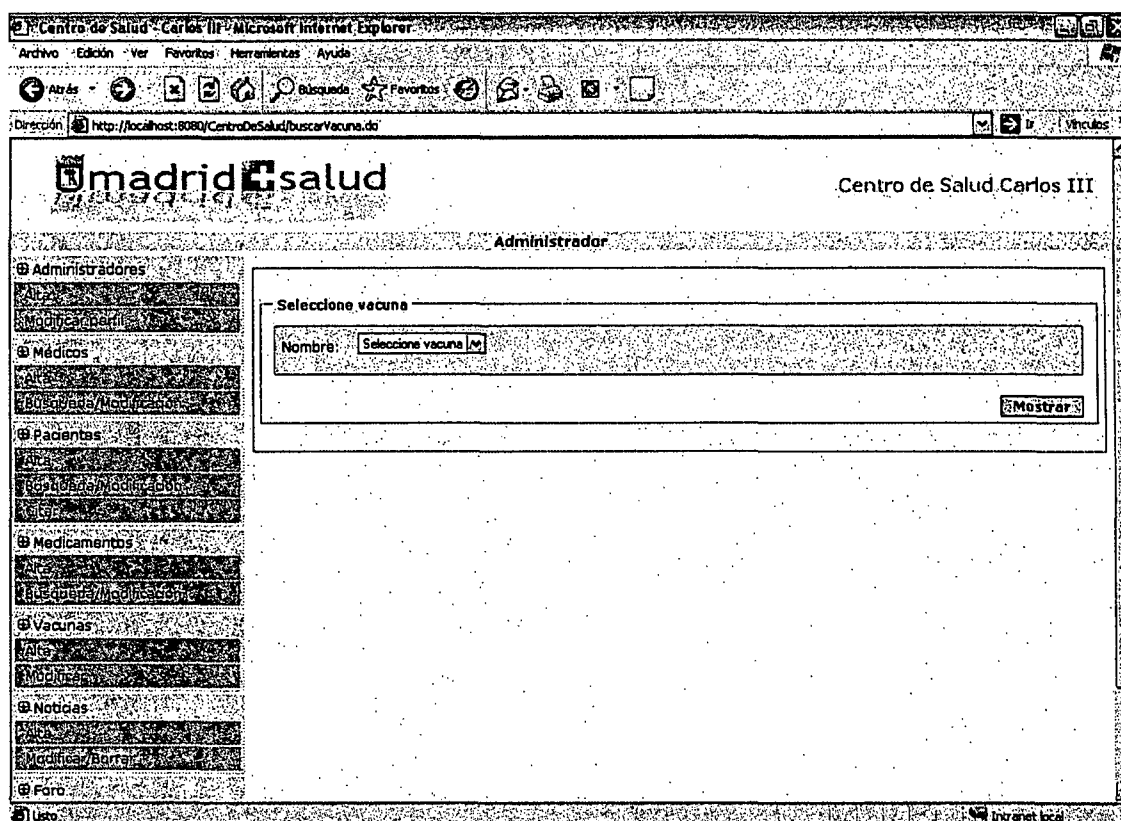
Una vez guardada, además de poder modificarla, existe la opción de eliminarla, pulsando el botón “Eliminar”.



- Figura 176: Vacuna guardada con éxito -

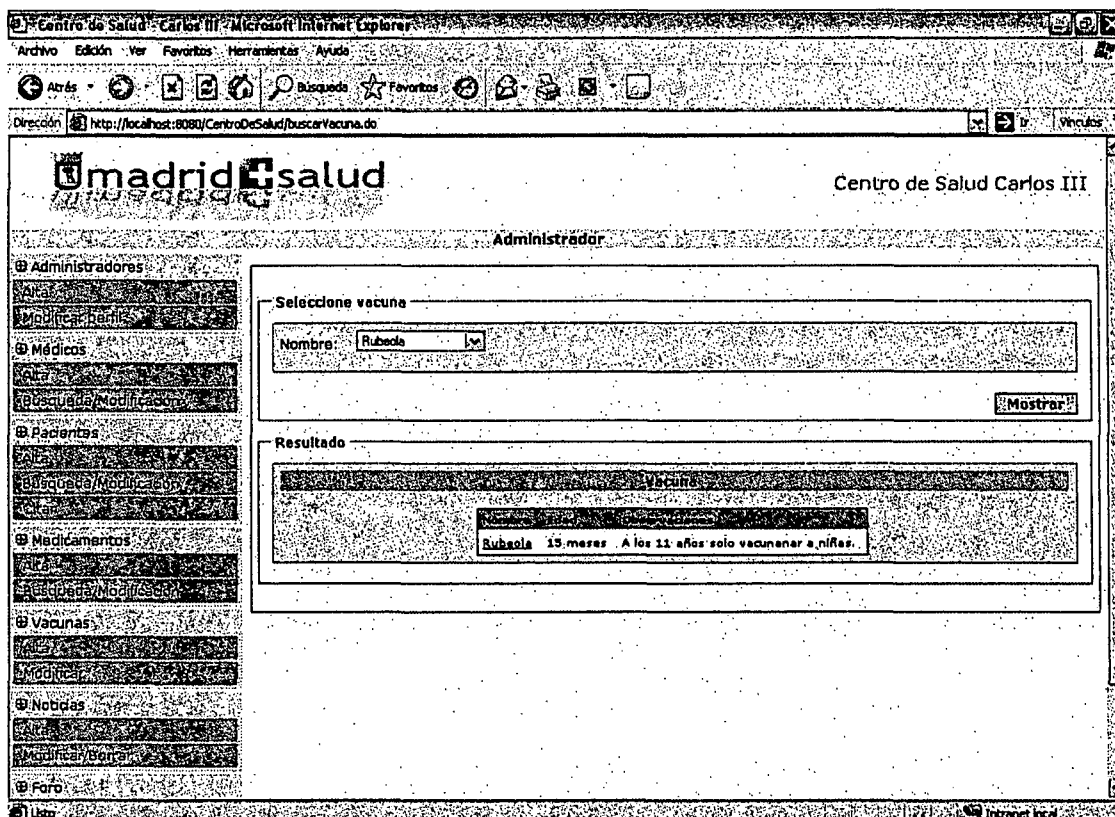
5.2.4.11.- MODIFICAR VACUNA

Cualquier administrador puede consultar las vacunas, ya sea para visualizar sus datos y/o actualizarlos y/o eliminarla. Para ello deberá pulsar en la opción del menú "Modificar" (Vacunas), apareciéndole por tanto la siguiente pantalla.



- Figura 177: Consultar vacuna -

En el combo desplegable, podrá seleccionar alguna de las vacunas que se encuentra almacenadas en la base de datos. Una vez seleccionada, debe pulsar “Mostrar” para que debajo aparezcan los datos de la vacuna.

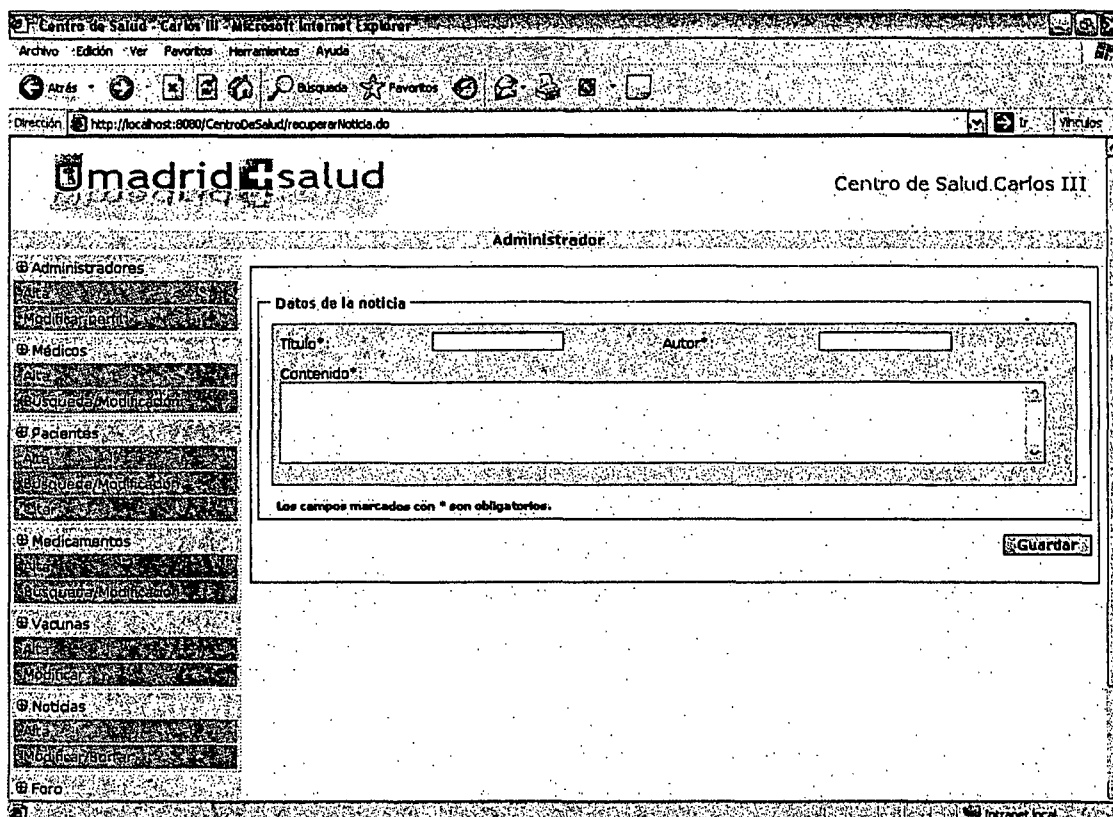


- Figura 178: Vacuna seleccionada -

Si desea modificar algún dato ó eliminar la vacuna de la base de datos, el administrador solo tendrá que pulsar sobre el nombre de la vacuna, apareciendo así sus datos con la opción de “Guardar” para modificarlos y la opción de “Eliminar”.

5.2.4.12.- DAR DE ALTA UNA NUEVA NOTICIA

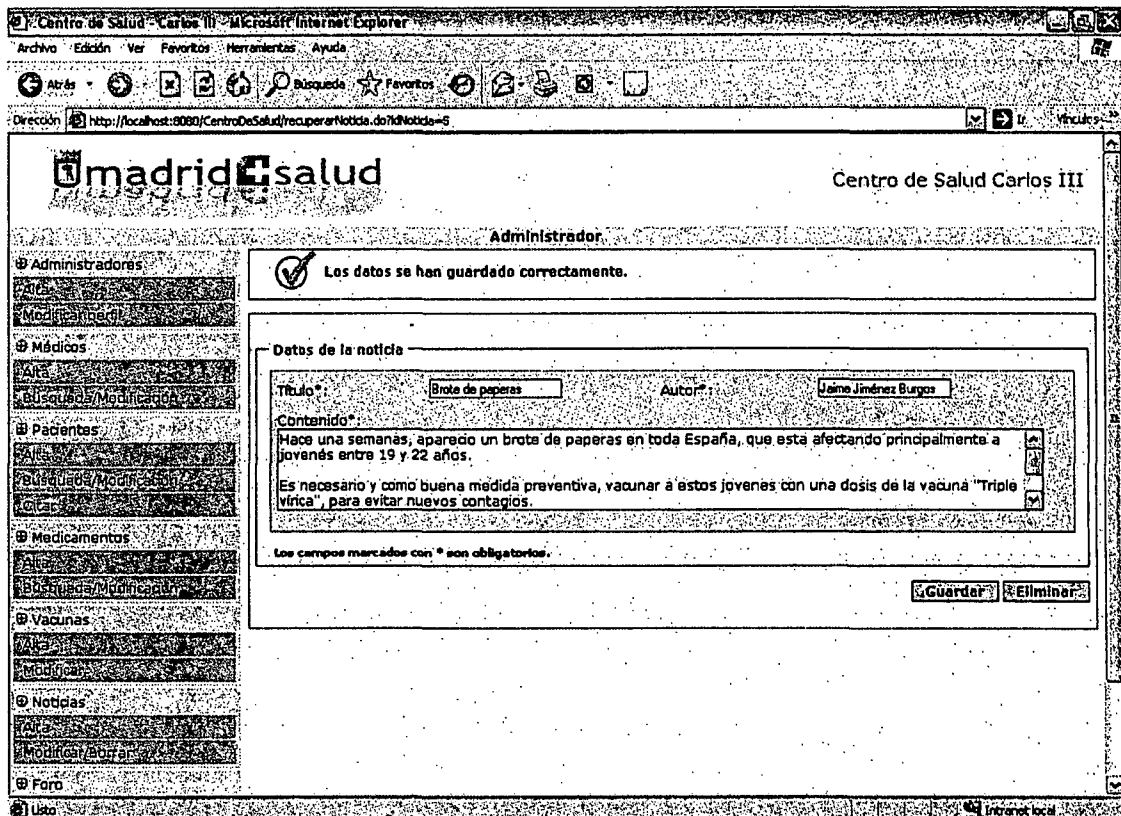
Para dar de alta una nueva noticia que el administrador haya recibido de un médico o haya encontrado interesante en una publicación médica, debe primeramente pulsar en la opción del menú “Alta” (Noticias). De esta forma le aparecerá una pantalla en la que se le pedirá que introduzca los datos de la noticia que va a dar de alta.



- Figura 179: Alta noticia -

Una vez introduzca los campos necesarios, el administrador pulsará sobre “Guardar”. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo en los datos introducidos, el administrador podrá modificarlos y tras pulsar de nuevo en “Guardar” estos se actualizarán.

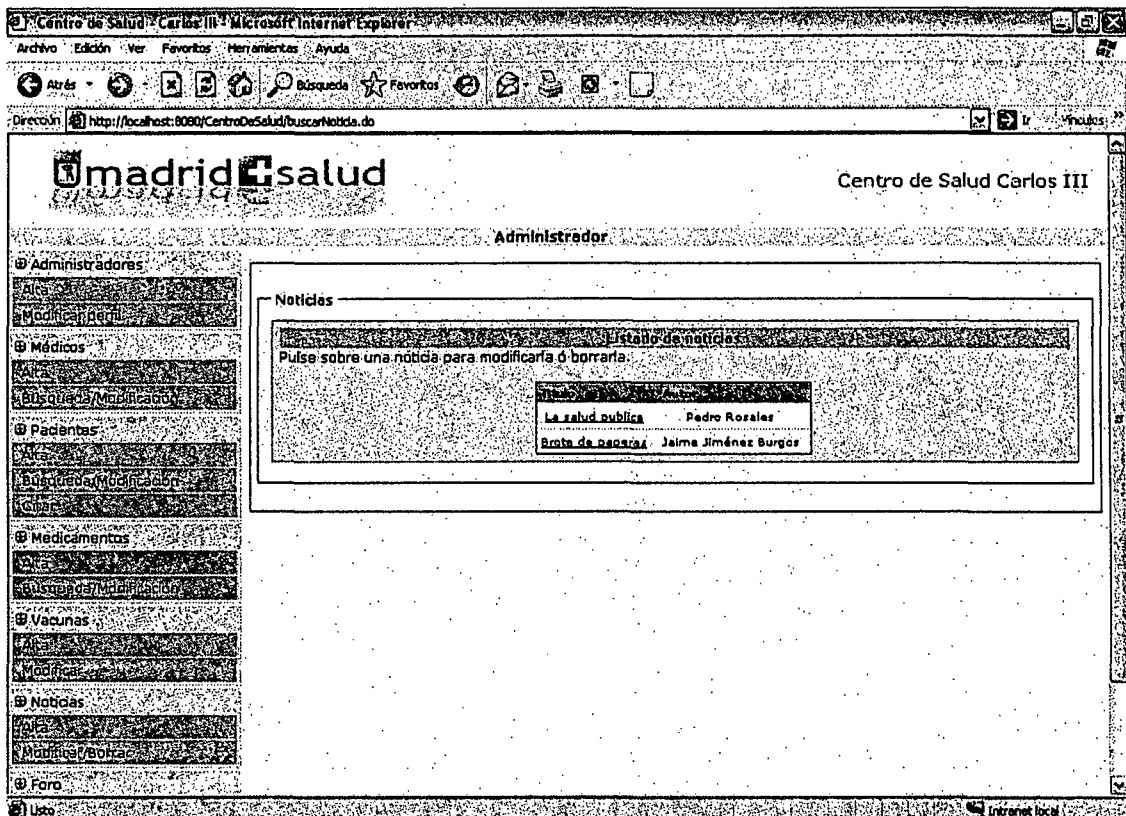
Una vez guardada, además de poder modificarla, existe la opción de eliminarla, pulsando el botón “Eliminar”.



- Figura 180: Noticia guardada con éxito -

5.2.4.13.- MODIFICAR NOTICIA

Cualquier administrador puede modificar y/o borrar una noticia de la base de datos. Para ello debe pulsar en la opción del menú “Modificar/Borrar” (Noticias), de tal forma que le aparecerá una pantalla con un listado que contendrá todas las noticias que existen.

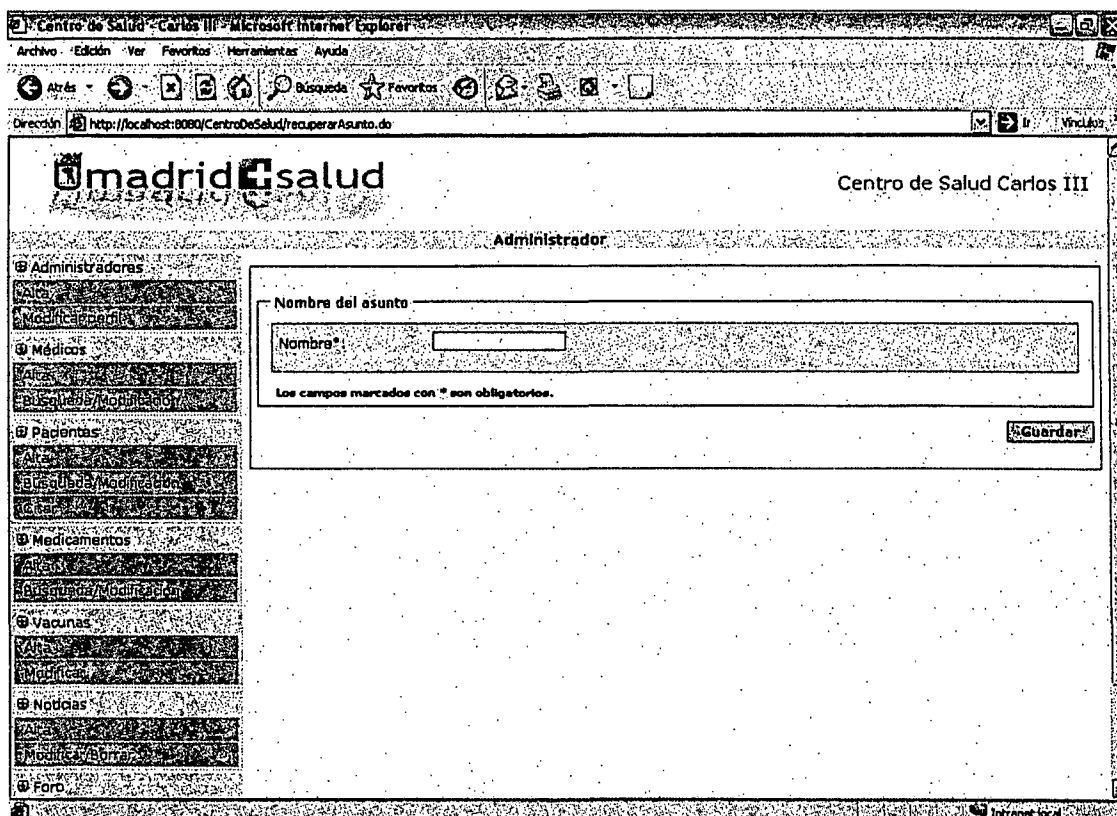


- Figura 181: Listado de noticias -

Pulsando sobre el título de una noticia, irá a una pantalla que le permitirá modificar cualquier dato y actualizar la noticia ó pulsar sobre “Eliminar” para borrarla definitivamente.

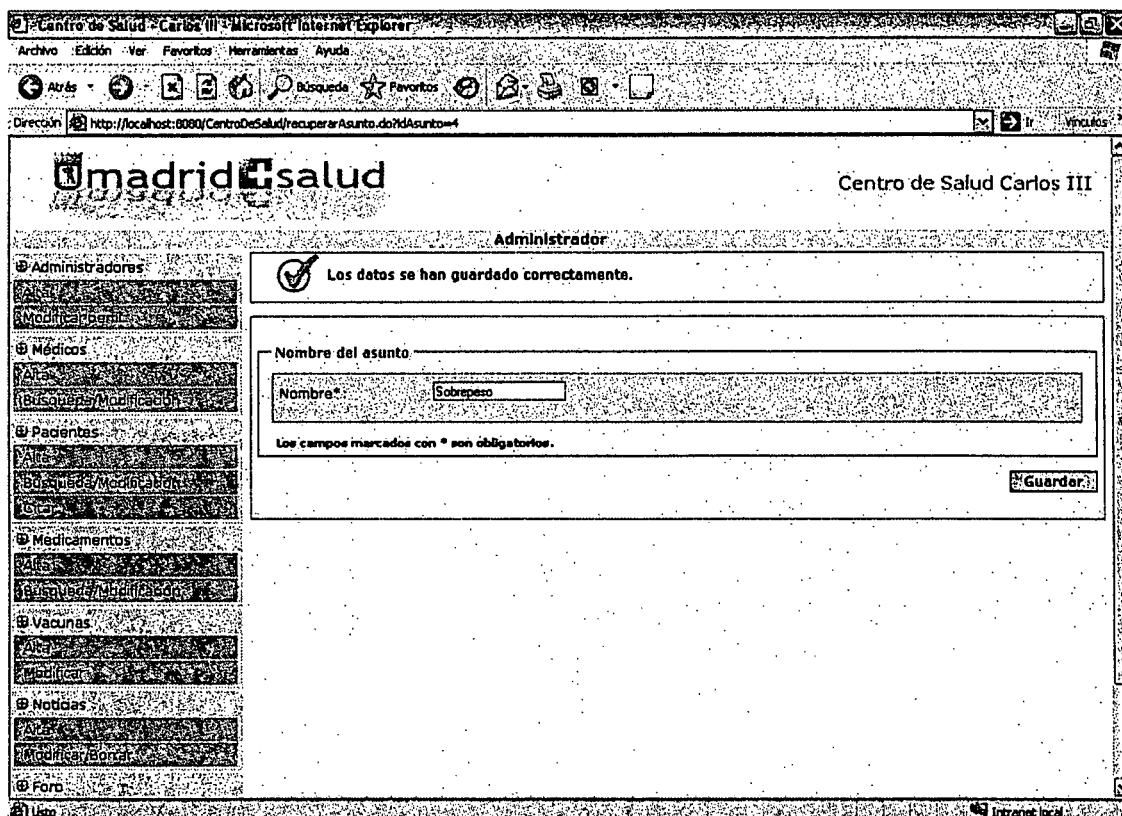
5.2.4.14.- DAR DE ALTA UN NUEVO ASUNTO DEL FORO

Dado que el administrador es el moderador del foro, este podrá dar de alta nuevos asuntos en el foro, pulsando sobre la opción “Alta asunto”



- Figura 182: Alta asunto -

Una vez pulsado, deberá introducir el nombre del nuevo asunto y pulsar en “Guardar”. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar el nombre del asunto para verificar que es correcto. En caso de encontrar algún fallo, el administrador podrá modificarlo y tras pulsar de nuevo en “Guardar” estos se actualizarán.

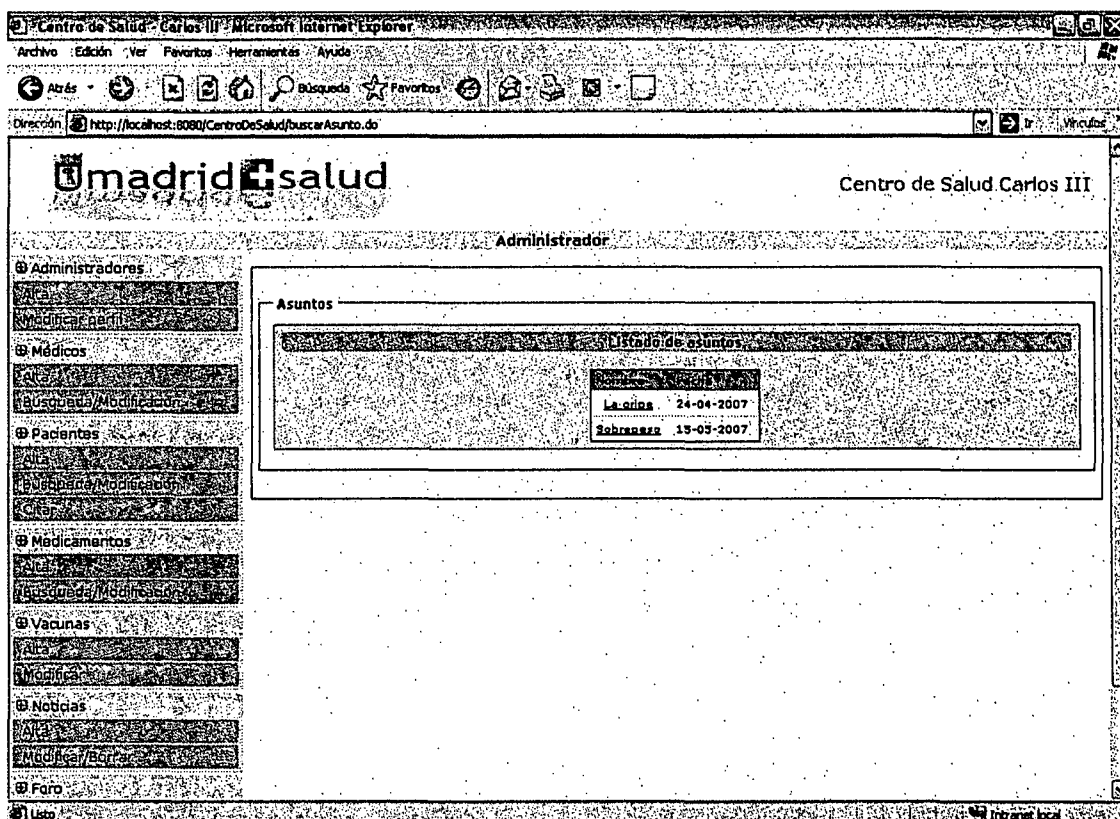


- Figura 183: Asunto guardado con éxito -

5.2.4.15.- BORRAR UN ASUNTO Ó PREGUNTA DEL FORO

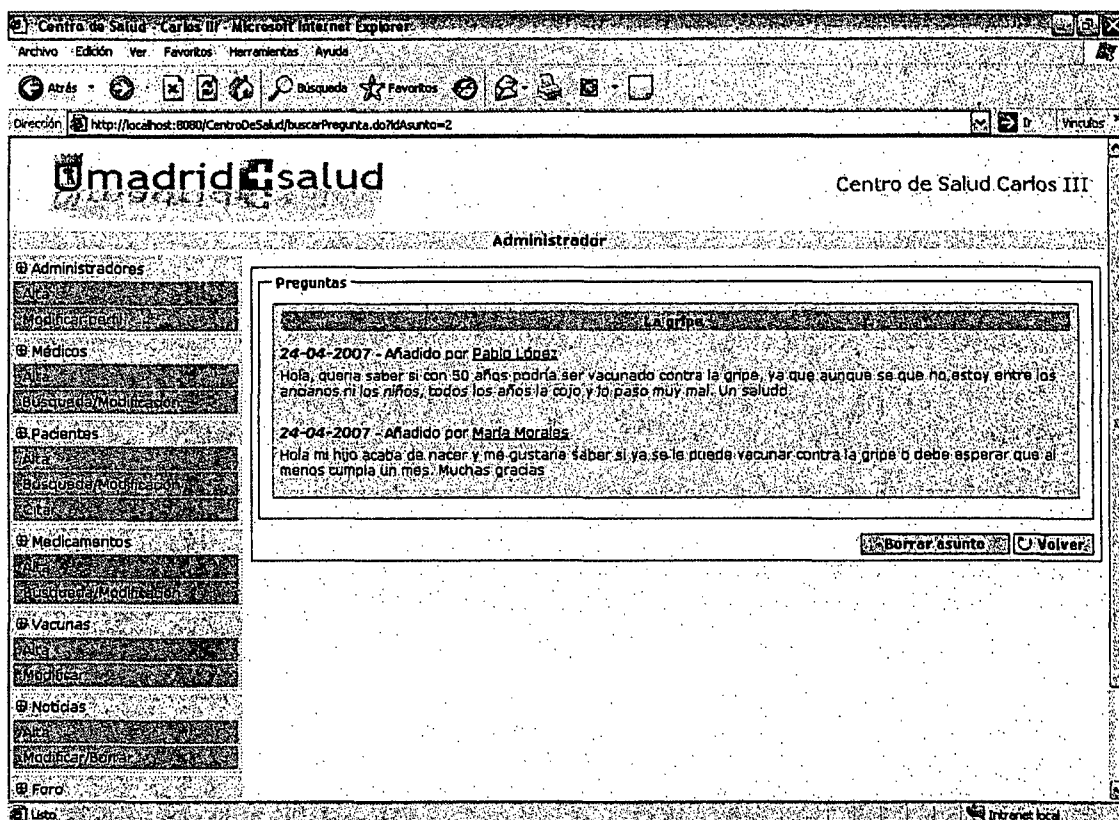
Como ya se ha comentado en el apartado anterior, dado que el administrador es el encargado de moderar el foro, este podrá eliminar cualquier asunto o pregunta que haga un usuario, ya sea por actualizar el foro, o porque existan palabras malsonantes en las preguntas realizadas, que ataquen a cualquier miembro del centro de salud.

Tras pulsar sobre la opción del menú "Borrar asunto/pregunta", aparecerá una pantalla con todos los asuntos del foro.



- Figura 184: Listado de asuntos -

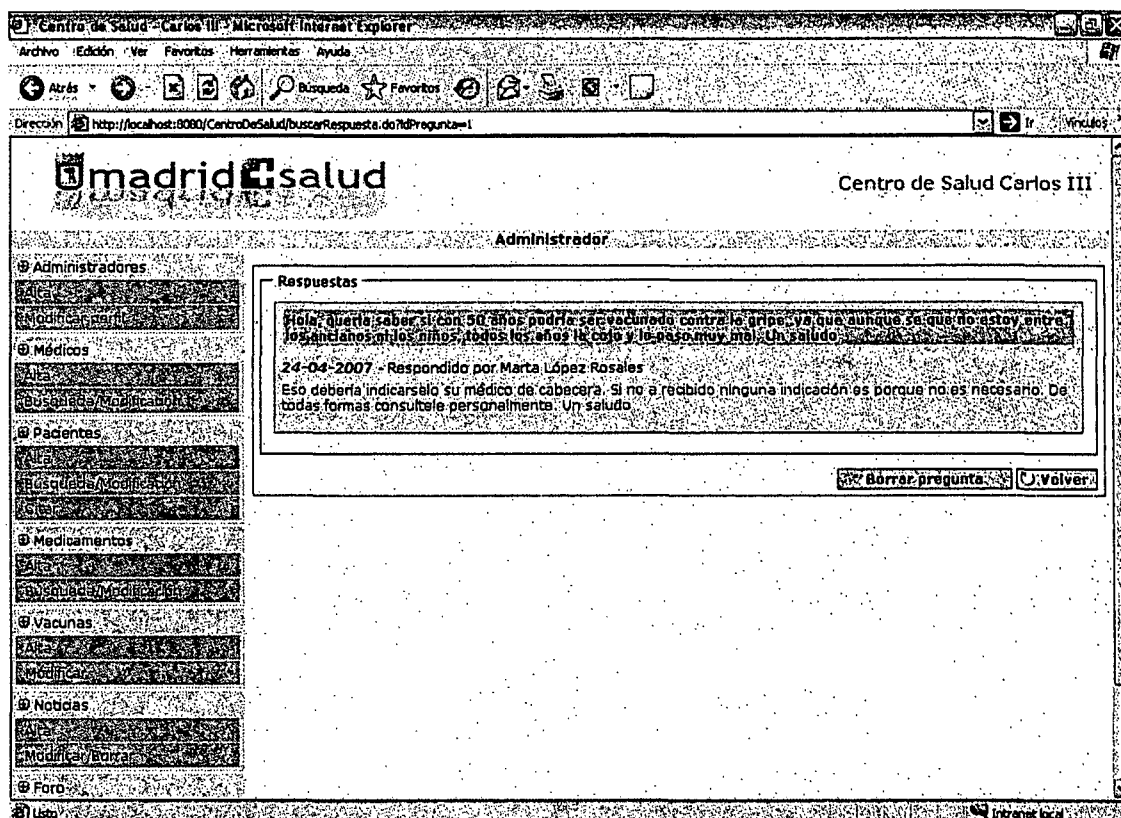
Si el administrador pulsa sobre el nombre de algún asunto, le aparecerá una pantalla con todas las preguntas pertenecientes a esos asuntos, realizadas por los usuarios, ó un mensaje de aviso, en caso de no existir ninguna pregunta.



- Figura 185: Listado de preguntas -

En esta pantalla, además existen tres opciones:

- Borrar asunto: Pulsando sobre ese botón desaparecerá el asunto del foro, desapareciendo por tanto todas las preguntas y respuestas pertenecientes a él.
- Volver: Tras pulsar en esta opción, se redirigirá a la pantalla anterior, es decir, al listado de asuntos de foro.
- El administrador puede pulsar sobre el nombre de la persona que ha introducido una pregunta, en caso de existir alguna, y visualizar así todas las respuestas que han dado los médicos del centro a esa pregunta, ó un mensaje de aviso, en caso de no existir ninguna respuesta.



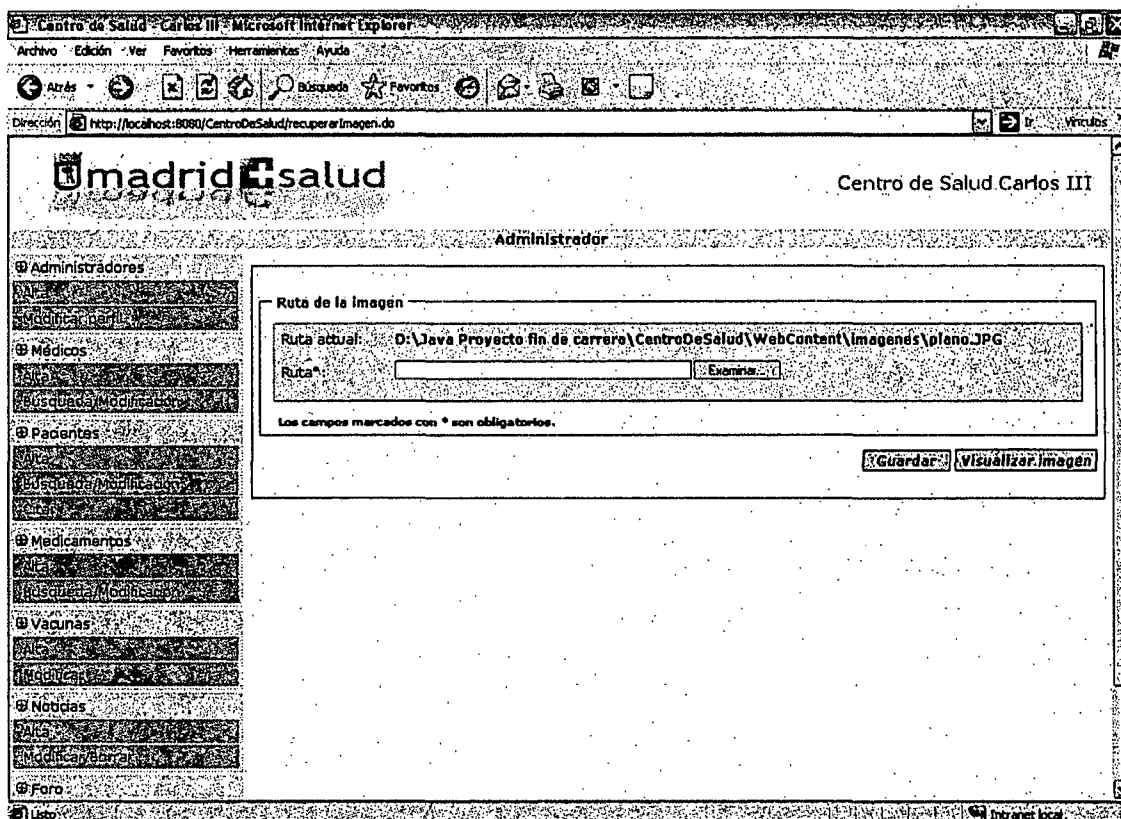
- Figura 186: Listado de respuestas -

En esta pantalla, además existen dos opciones:

- Borrar pregunta: Pulsando sobre ese botón desaparecerá la pregunta del foro, desapareciendo por tanto todas sus respuestas.
- Volver: Tras pulsar en esta opción, se redirigirá a la pantalla anterior, es decir, al listado de preguntas.

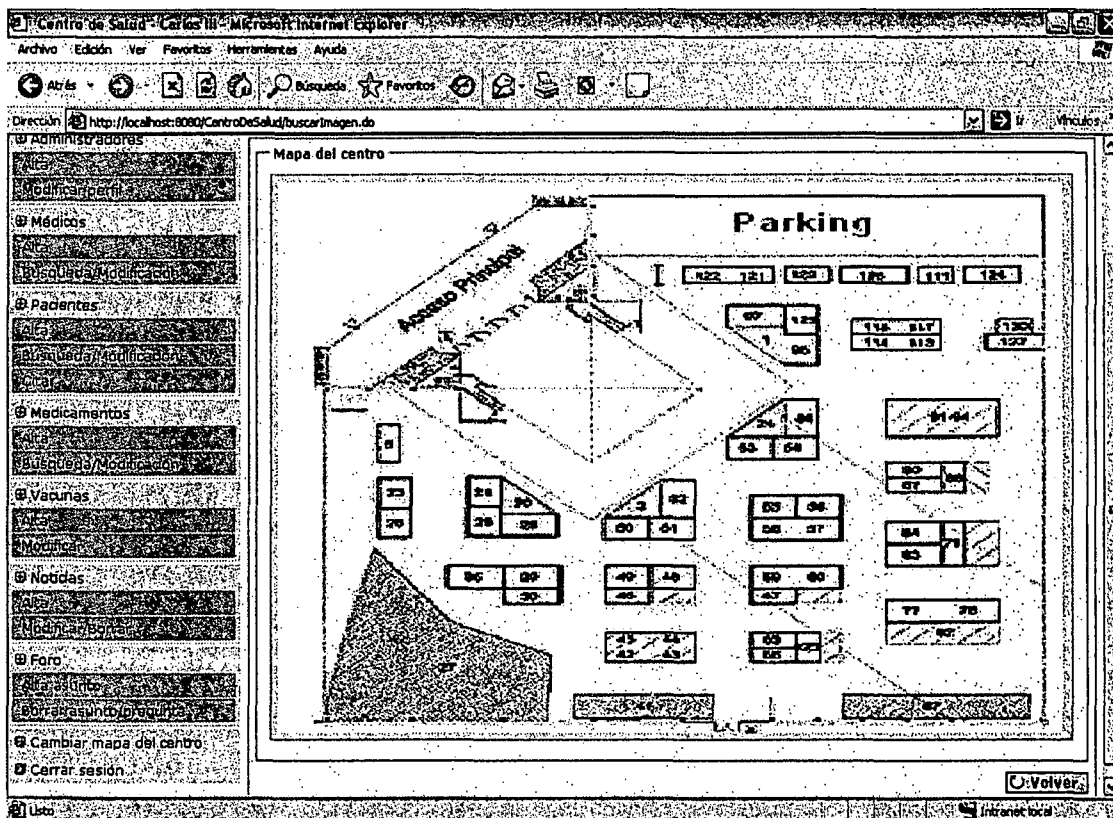
5.2.4.16.- CAMBIAR MAPA DEL CENTRO

Cualquier administrador, podrá cambiar el mapa del centro. Para ello debe pulsar en la opción del menú "Cambiar mapa del centro". Una vez pulsada, aparecerá la siguiente página.



- Figura 187: Cambiar mapa -

En esta pantalla, si no existe ninguna imagen introducida con anterioridad, solo aparecerá el campo para introducir la ruta y el botón "Guardar". En cambio, si existe una imagen ya, además de poder cambiarla, antes podrá visualizarla, pulsando sobre "Visualizar imagen".



- Figura 188: Visualizar mapa -

Si el administrador decide visualizar el mapa, podrá pulsar en el botón “Volver”, para regresar a la página anterior.

5.2.5.- OPCIONES DEL MÉDICO

5.2.5.1.- CONSULTAR PACIENTE

Cualquier médico puede consultar los pacientes que hay en el centro de salud, ya sea para visualizar sus datos y/o actualizarlos. Para ello deberá pulsar en la opción del menú “Búsqueda/Modificación” (Pacientes).

En la pantalla que aparecerá se presentan dos opciones:

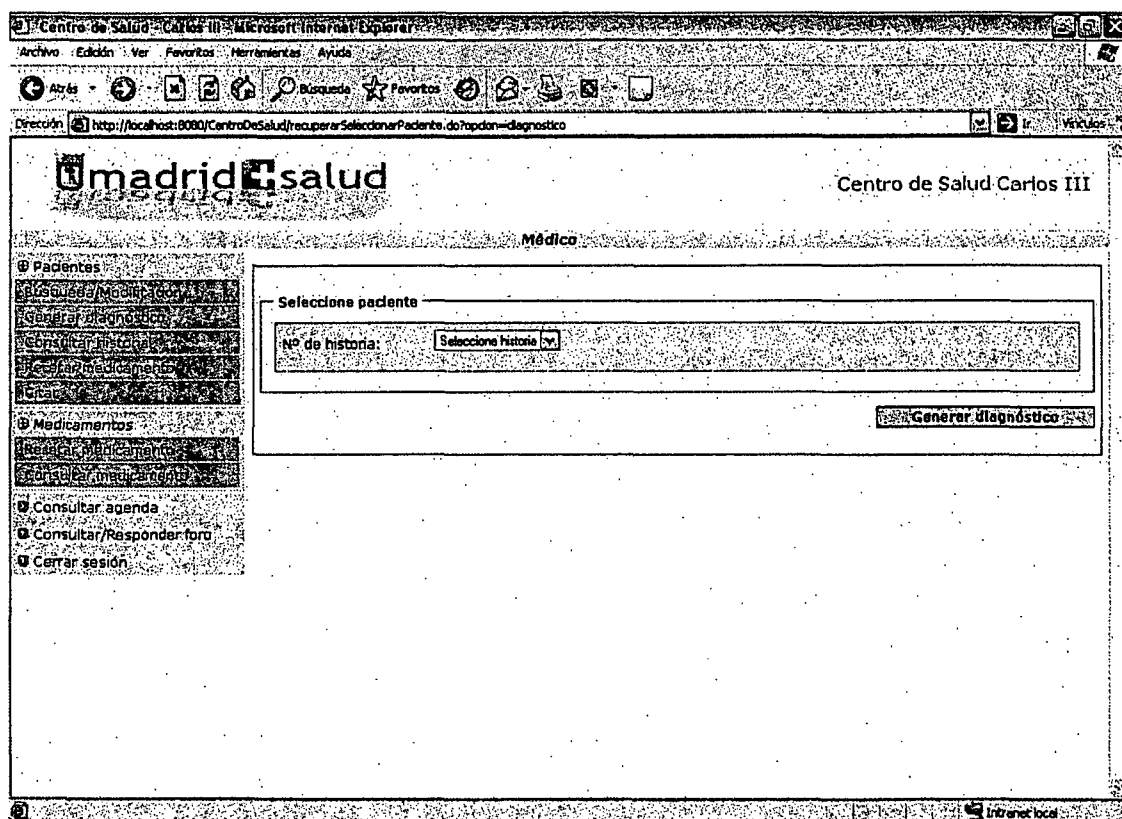
- Buscar a un paciente introduciendo algún/os de los criterios de búsqueda (DNI, nombre...) y pulsar buscar, de forma que se muestren todos los pacientes que coincidan con los criterios establecidos.
- Pulsar sobre el botón “Mostrar todos” para que aparezcan todos los pacientes del centro.

En ambas opciones, si no existen resultados que mostrar, aparecerá un mensaje de aviso.

Si tras la búsqueda el médico desea ver los datos de un paciente, solo deberá pulsar sobre el número de historia del paciente que desea consultar, apareciendo así una pantalla con sus datos, en la cual podrá modificar los datos necesarios y guardarlos pulsando sobre el botón “Guardar”, además de poder añadir nuevas alergias.

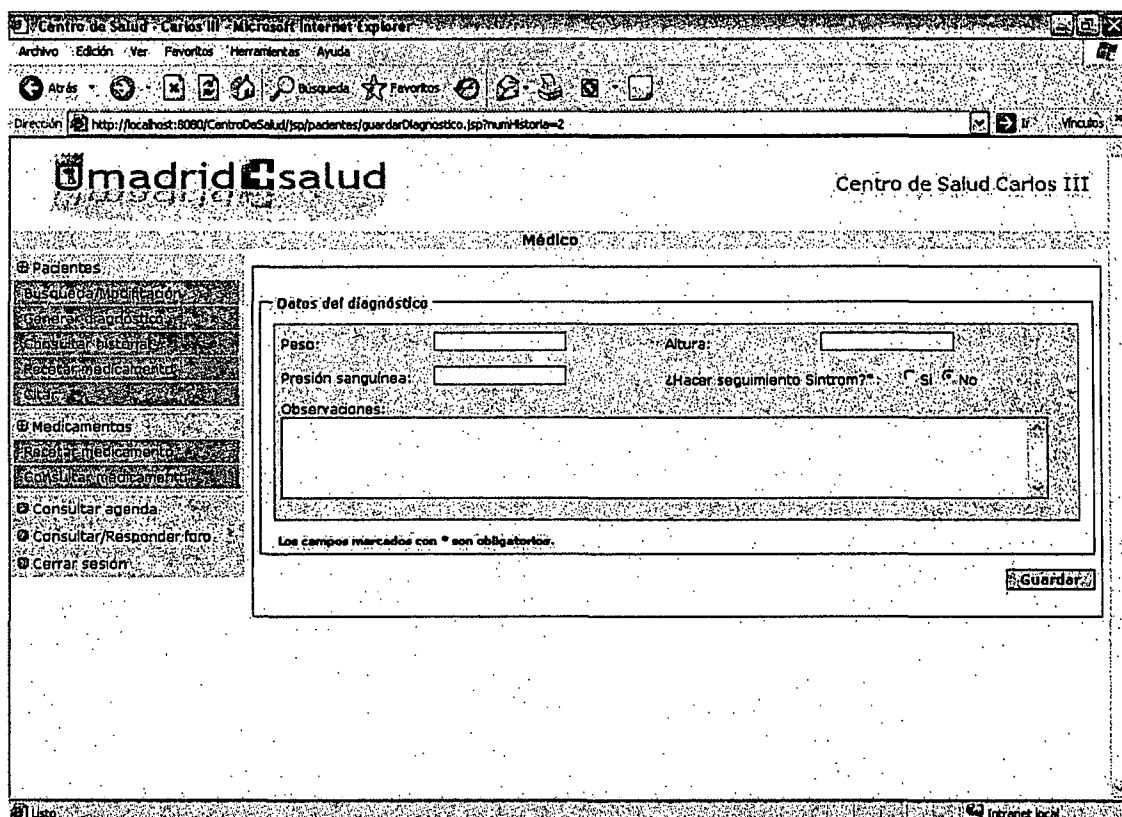
5.2.5.2.- GENERAR DIAGNÓSTICO

Cualquier médico podrá generar diagnósticos de los pacientes que tiene asignados. Para ello, deberá pulsar sobre la opción del menú “Generar diagnóstico”, apareciéndole la siguiente pantalla.



- Figura 189: Generar diagnóstico -

Aquí el médico podrá seleccionar uno de sus pacientes a través del combo desplegable y tras pulsar el botón “Generar diagnóstico” comenzar a rellenar los campos necesarios.



Centro de Salud Carlos III

Madrid Salud

Médico

Pacientes

- Busqueda/Modificación
- General diagnóstico
- Consultar historia
- Recetar medicamento
- Alta

Medicamentos

- Recetar medicamento
- Consultar medicamento

Consultar agenda

- Consultar/Responder foro
- Cerrar sesión

Datos del diagnóstico

Peso: Altura:

Presión sanguínea: ¿Hacer seguimiento Sintrom? ☐ Si ☐ No

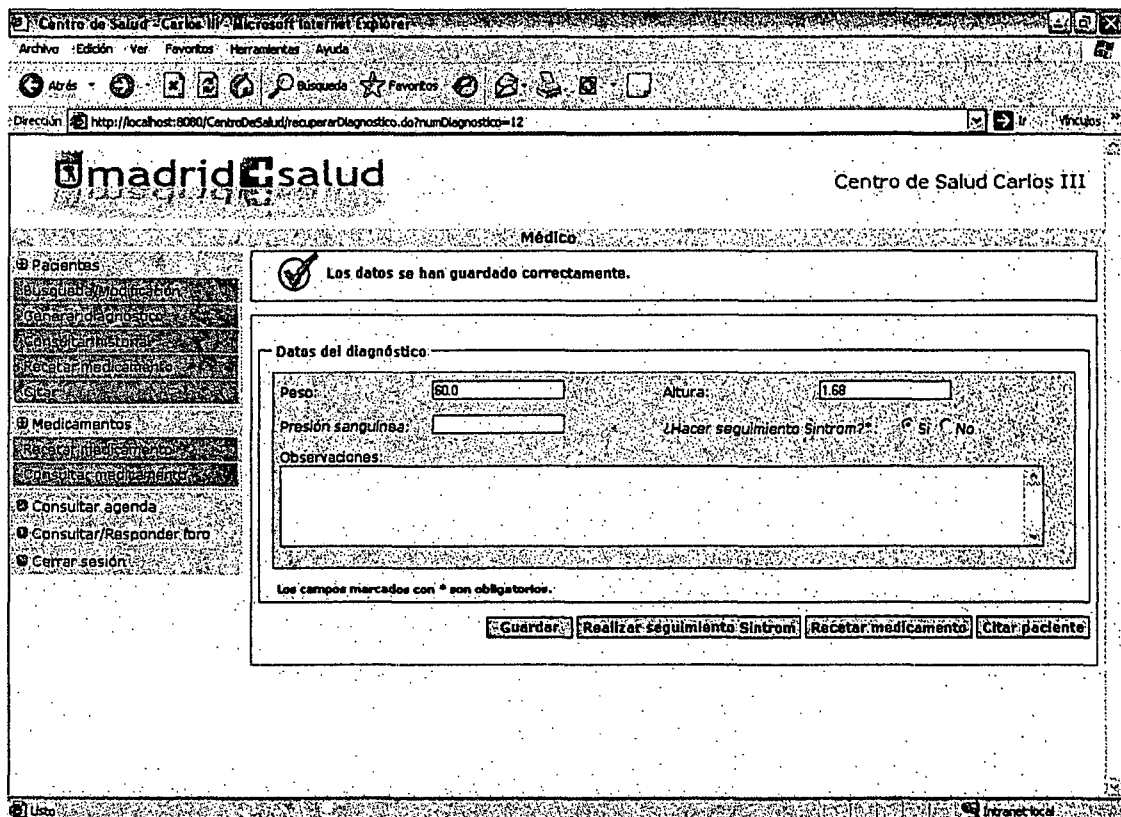
Observaciones:

Los campos marcados con * son obligatorios.

Guardar

- Figura 190: Datos diagnóstico -

Una vez introduzca los datos necesarios, el médico pulsará sobre “Guardar”. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo en los datos introducidos, el médico podrá modificarlos y tras pulsar de nuevo en “Guardar” estos se actualizarán.



The screenshot shows a web browser window with the URL <http://localhost:8080/CentroDeSalud/recuperarDiagnostico.do?munDiagnostico=12>. The page header includes the "madrid salud" logo and "Centro de Salud Carlos III". A navigation menu on the left lists options like "Pacientes", "Medicamentos", "Consultar agenda", and "Cerrar sesión". The main content area, titled "Médico", displays a success message: "Los datos se han guardado correctamente." Below this, a form titled "Datos del diagnóstico:" contains fields for "Peso" (60.0), "Altura" (1.68), "Presión sanguínea", and "Hacer seguimiento Sintrom?". There is also a large text area for "Observaciones". At the bottom, a note states "Los campos marcados con * son obligatorios." and four buttons are visible: "Guardar", "Realizar seguimiento Sintrom", "Recetar medicamento", and "Citar paciente".

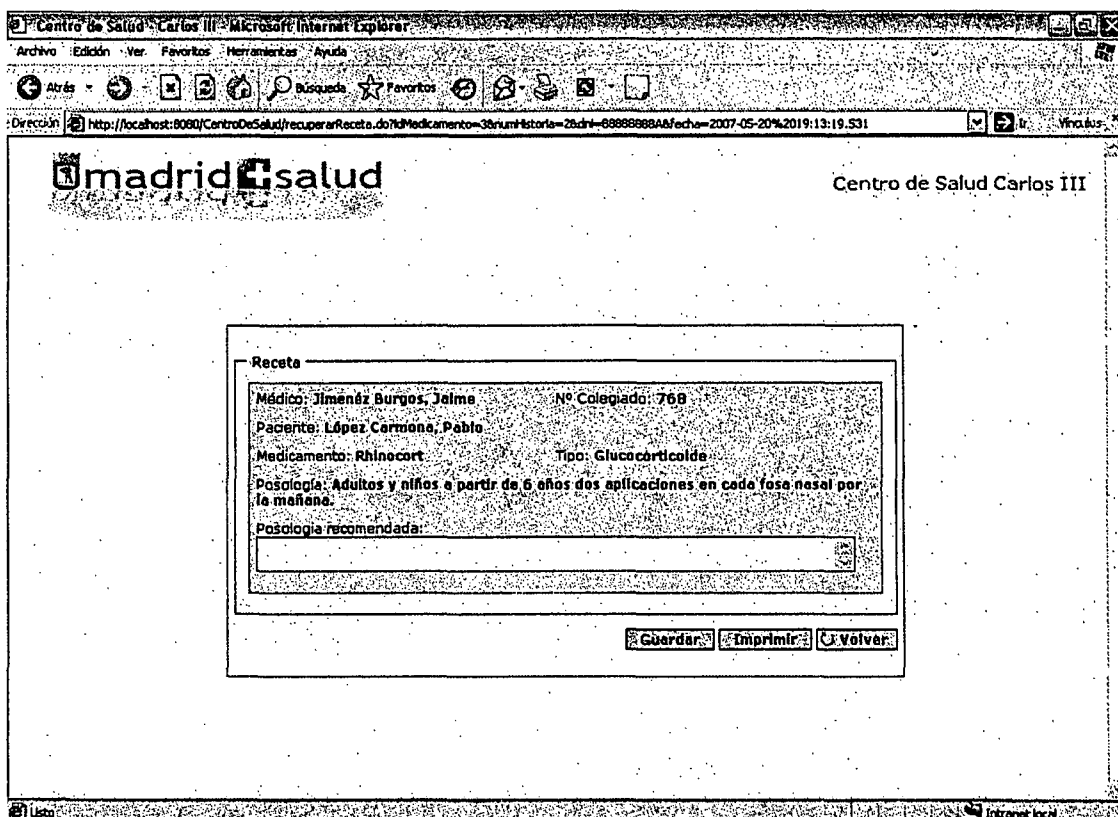
- Figura 191: Diagnóstico guardado con éxito -

Como se puede observar en la imagen anterior, tras guardar el diagnóstico de un paciente, aparecerán nuevos botones:

- Recetar medicamento: Se puede recetar uno o varios medicamentos a un paciente. Tras pulsar sobre el botón, aparecerá una pantalla, que funciona de la misma manera que "Consultar medicamento" (ver apartado 5.2.4.9).

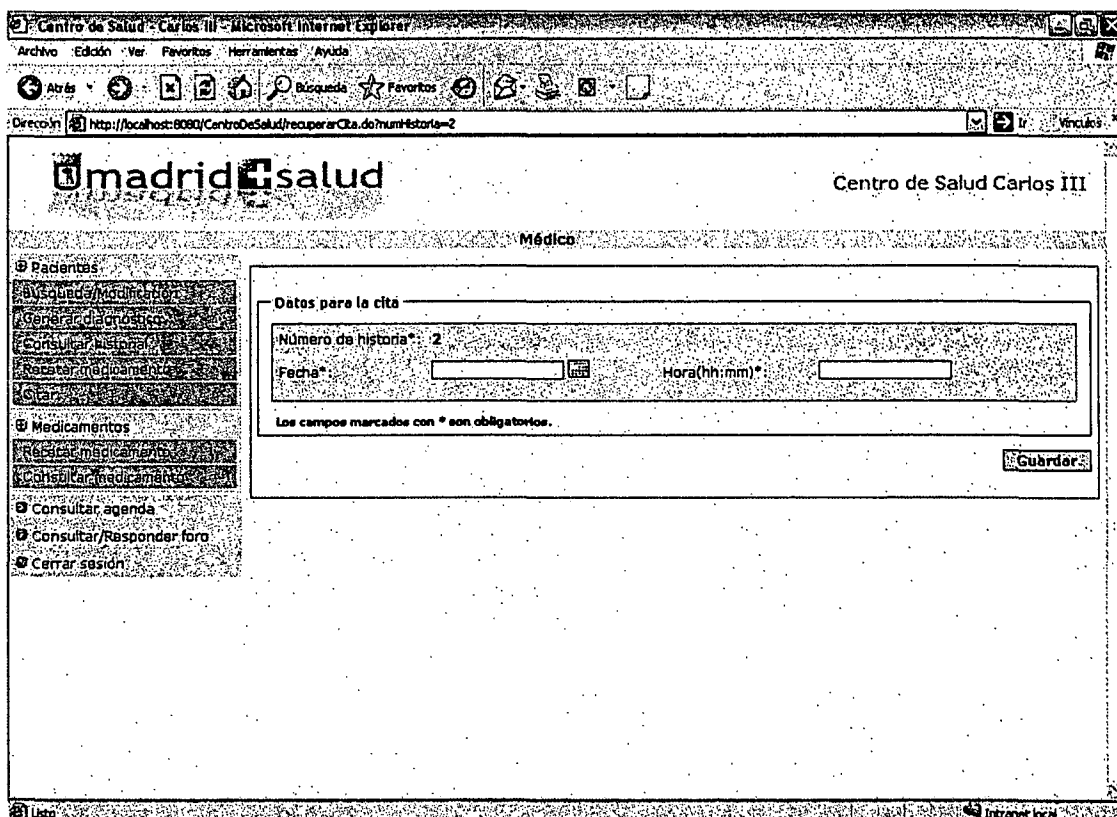
Tras buscar el medicamento que se desea recetar y pulsar sobre el nombre de este, se generará automáticamente la receta.

Cuando aparezca la receta se dará opción a modificar la posología del medicamento de forma manual de manera que tras pulsar en "Guardar" esta se quede almacenada; se podrá pulsar sobre "Imprimir" para poder dar la receta al paciente; y por último se deberá pulsar sobre el botón "Volver" para regresar a la pantalla anterior y recetar más medicamentos al paciente ó terminar.



- Figura 192: Receta -

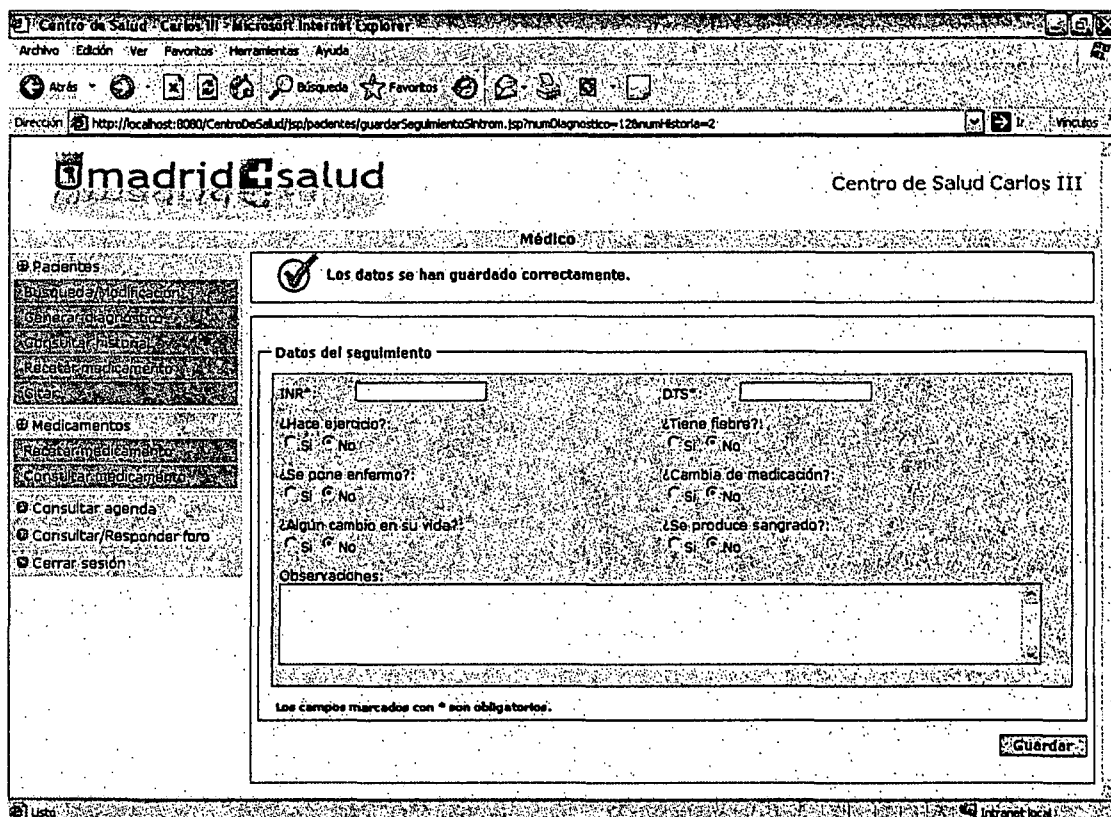
- Citar paciente: El médico podrá citar al paciente. Tras pulsar sobre este botón., aparecerá una pantalla en la que deberá introducir únicamente la fecha y hora de la cita. (ver apartado 5.2.4.7).



The screenshot shows a web browser window titled 'Centro de Salud Carlos III - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8080/CentroDeSalud/recuperarCita.do?numHistoria=2'. The page header includes the 'madridsalud' logo and 'Centro de Salud Carlos III'. The main content area is titled 'Médico' and contains a 'Datos para la cita' form. The form has two input fields: 'Número de historia*' with the value '2' and 'Fecha*' with a calendar icon. To the right, there is a 'Hora(hh:mm)*' field with a time selection icon. Below the form, a message states 'Los campos marcados con * son obligatorios.' and a 'Guardar' button is visible. A left sidebar menu lists various functions under 'Pacientes' and 'Medicamentos'.

- Figura 193: Médico cita a paciente -

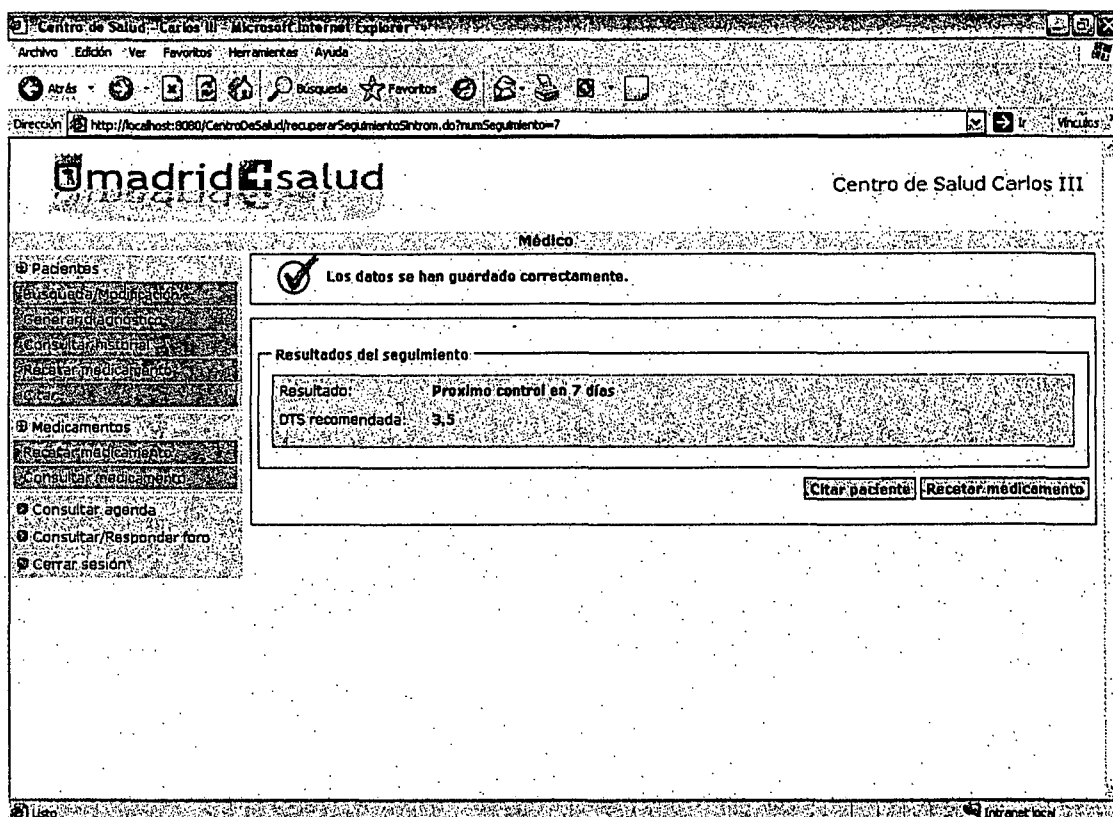
- Realizar seguimiento de Sintrom: Este botón solo aparecerá en el caso de que se haya marcado en el diagnóstico que se va a realizar el seguimiento de Sintrom. Si se pulsa sobre él, aparecerá la siguiente pantalla.



- Figura 194: Seguimiento Sintrom -

En esta pantalla el médico deberá rellenar los datos pertinentes y pulsar en “Guardar”. En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje de éxito y se indicará el resultado y la dosis recomendada para ese paciente. Además se dará opción a recetar un medicamento y/o citar al paciente.

Si la DTS que calcula el seguimiento es menor que 0.5 ó mayor que 30, se dará opción a que el médico la corrija manualmente y tras pulsar “Guardar” se refresquen los resultados.



Centro de Salud Carlos III

Médico

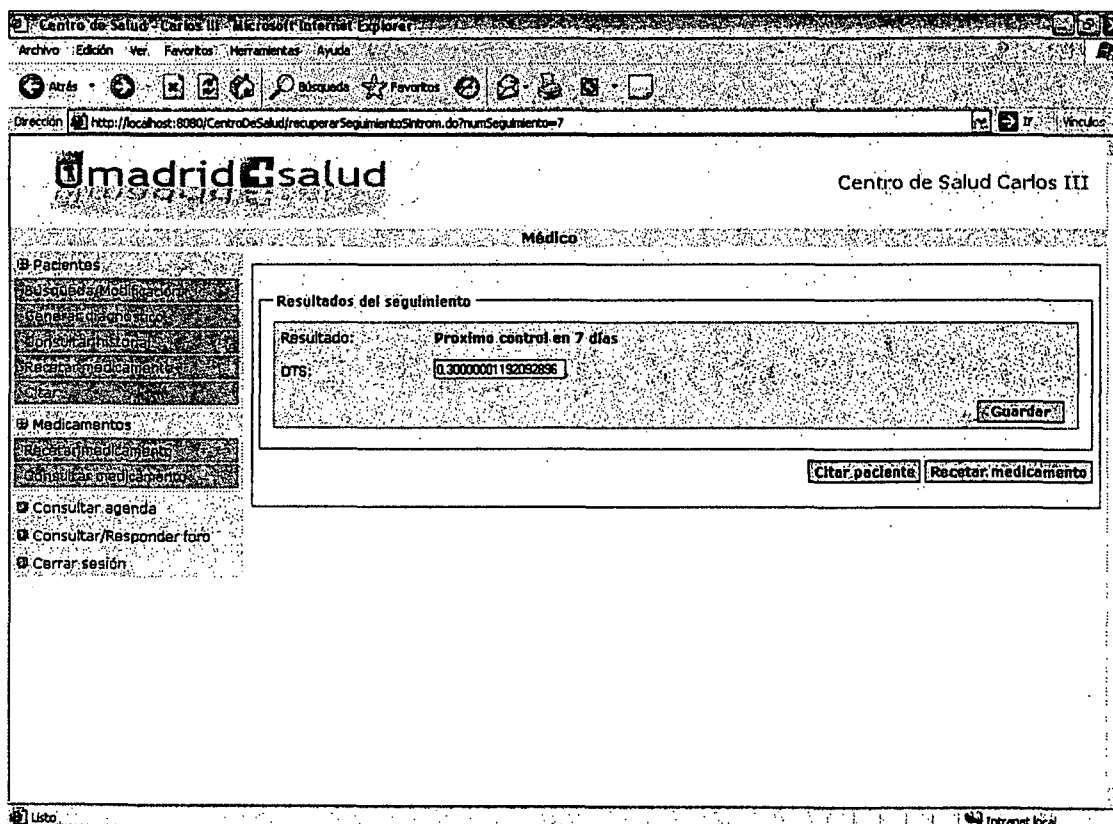
Los datos se han guardado correctamente.

Resultados del seguimiento

Resultado:	Proximo control en 7 días
DTS recomendada:	3,5

[Citar paciente](#) [Recetar medicamento](#)

- Figura 195: Seguimiento Sintrom guardado con éxito -



Centro de Salud Carlos III

Médico

Resultados del seguimiento

Resultado:	Proximo control en 7 días
DTS:	0.30000001192092896

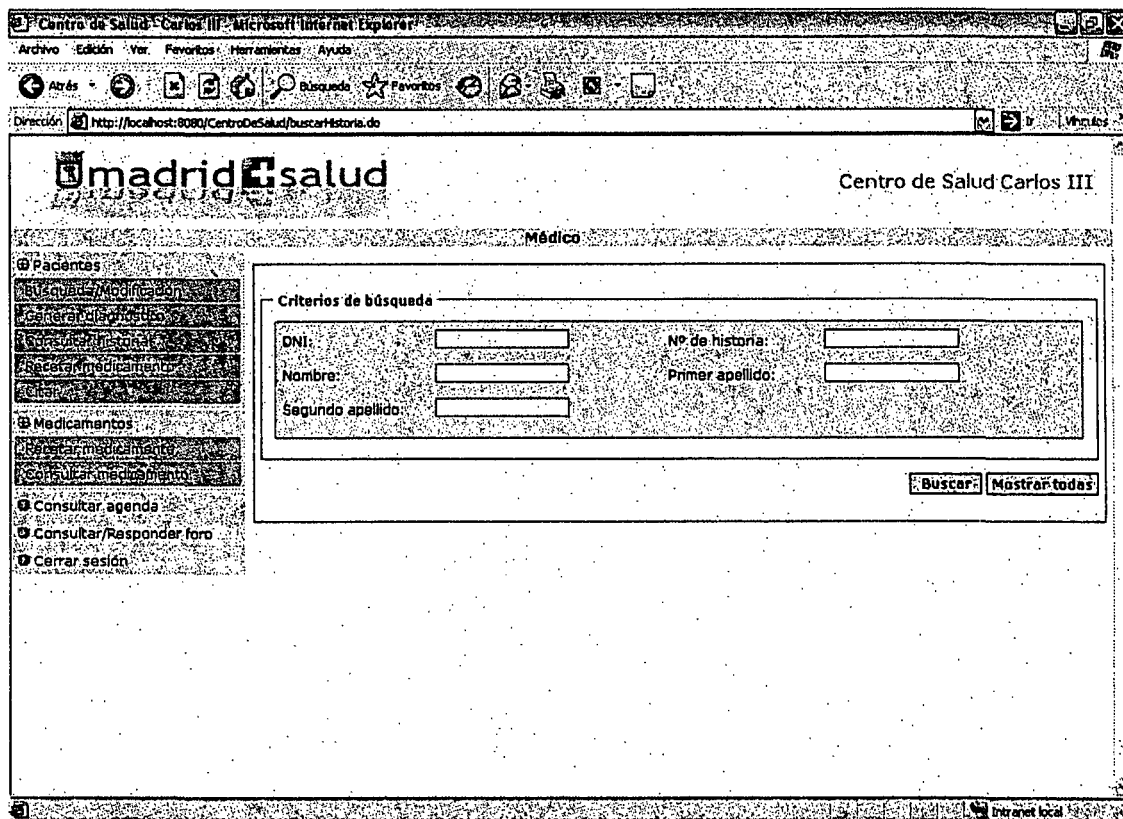
[Guardar](#)

[Citar paciente](#) [Recetar medicamento](#)

- Figura 196: DTS < 0,5 -

5.2.5.3.- CONSULTAR HISTORIAL

Cualquier médico puede consultar el historial de cualquier paciente del centro de salud. Para ello debe pulsar en la opción “Consultar historial” apareciéndole la siguiente pantalla.

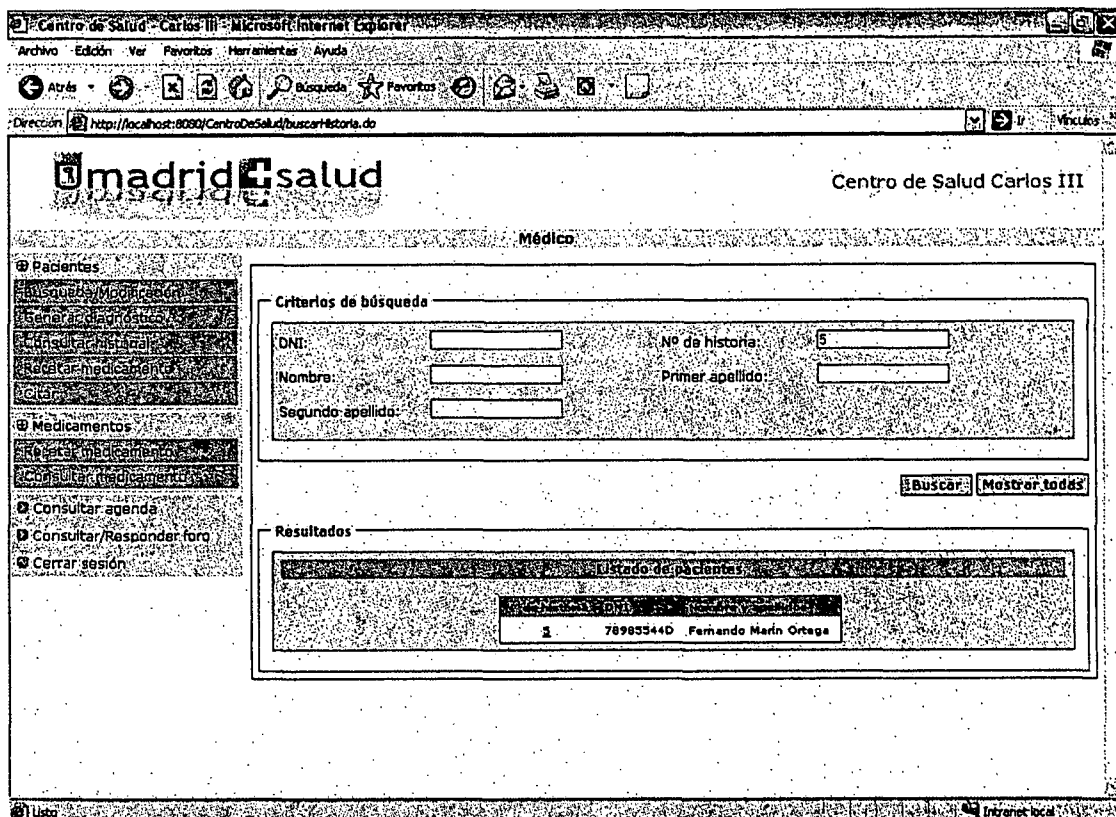


- Figura 197: Consultar historial -

En la pantalla se presentan dos opciones:

- Buscar el historial de un paciente introduciendo algún/os de los criterios de búsqueda (DNI, nombre...) y pulsar buscar, de forma que se muestren todos los historiales que coincidan con los criterios establecidos.
- Pulsar sobre el botón “Mostrar todos” para que aparezcan todos los historiales del centro.

En ambas opciones, si no existen resultados que mostrar, aparecerá un mensaje de aviso.



Centro de Salud Carlos III

Médico

Criterios de búsqueda

DNI: Nº de historia:

Nombre: Primer apellido:

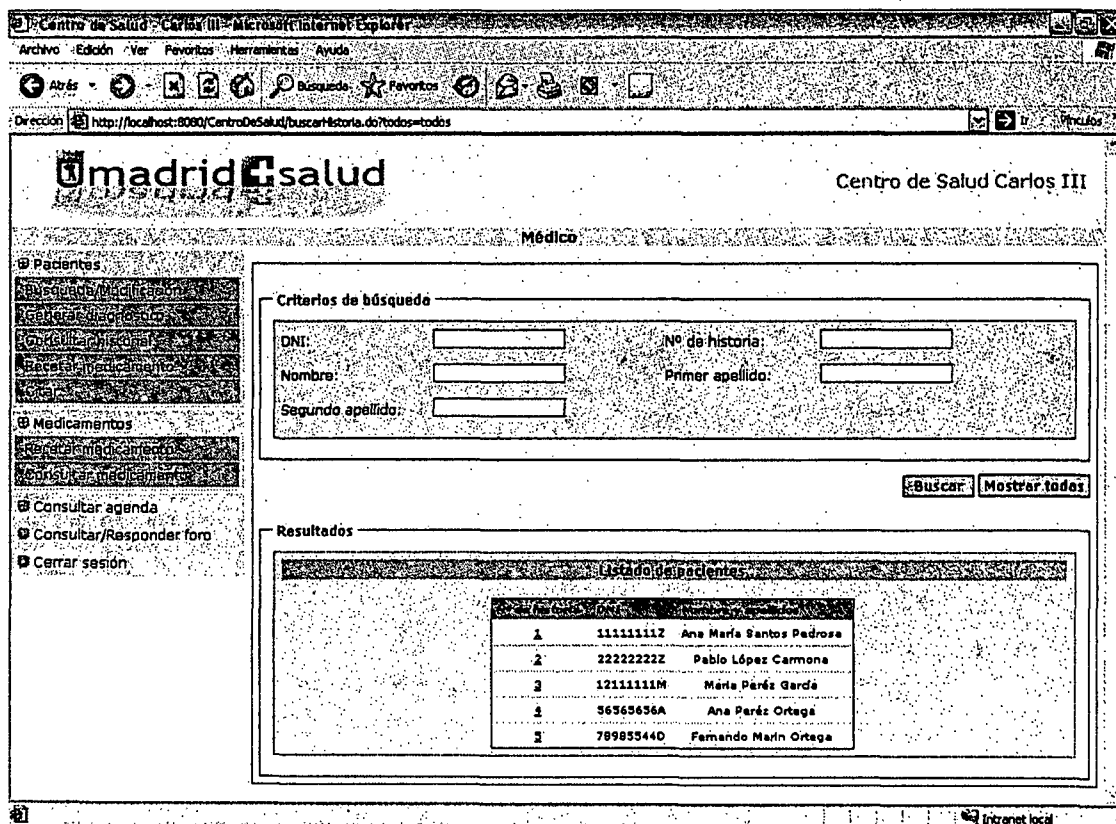
Segundo apellido:

Resultados

Estado de pacientes

Nº	DNI	Nombre
1	78985544D	Fernando Marín Ortega

- Figura 198: Consultar historial introduciendo algún criterio -



Centro de Salud Carlos III

Médico

Criterios de búsqueda

DNI: Nº de historia:

Nombre: Primer apellido:

Segundo apellido:

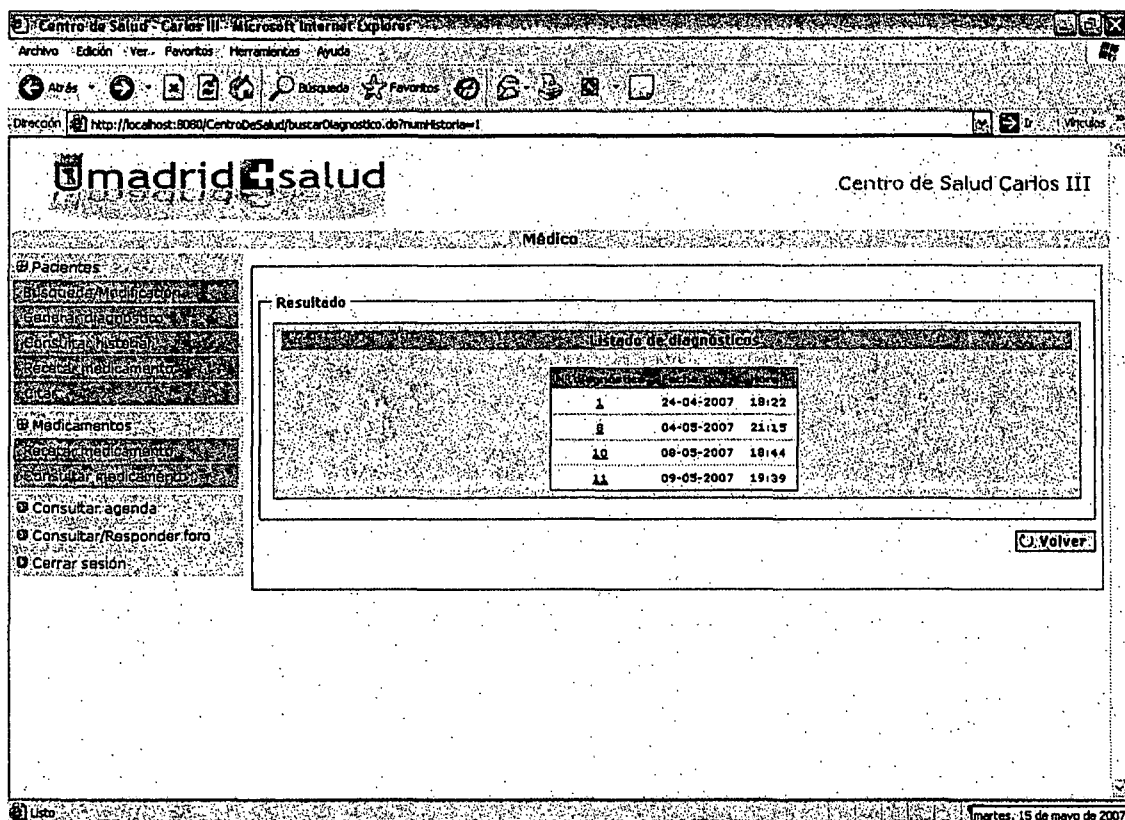
Resultados

Estado de pacientes

Nº	DNI	Nombre
1	111111112	Ana María Santos Pedrosa
2	222222222	Pablo López Carmona
3	12111111M	María Pérez García
4	56565656A	Ana Pérez Ortega
5	78985544D	Fernando Marín Ortega

- Figura 199: Mostrar todos los historiales -

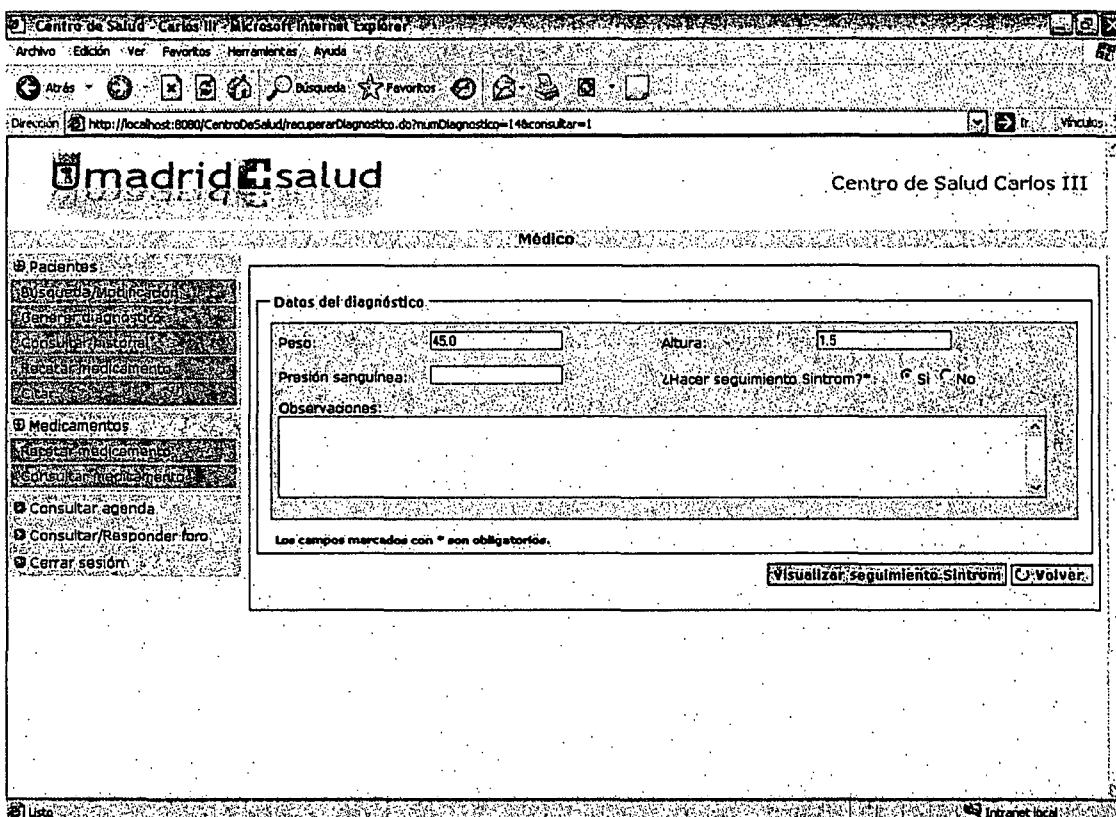
Si tras la búsqueda el médico desea ver los diagnósticos de un paciente, solo deberá pulsar sobre el número de historia del historial que desea consultar, apareciendo así una pantalla con el listado de diagnósticos ó en caso de no existir ninguno un mensaje de aviso.



- Figura 200: Listado de diagnósticos -

En esta pantalla existen dos opciones:

- Pulsar en el botón “Volver” y regresar a la pantalla anterior.
- Pulsar sobre el número de un diagnóstico, en caso de existir alguno, y consultar los datos de ese diagnóstico.

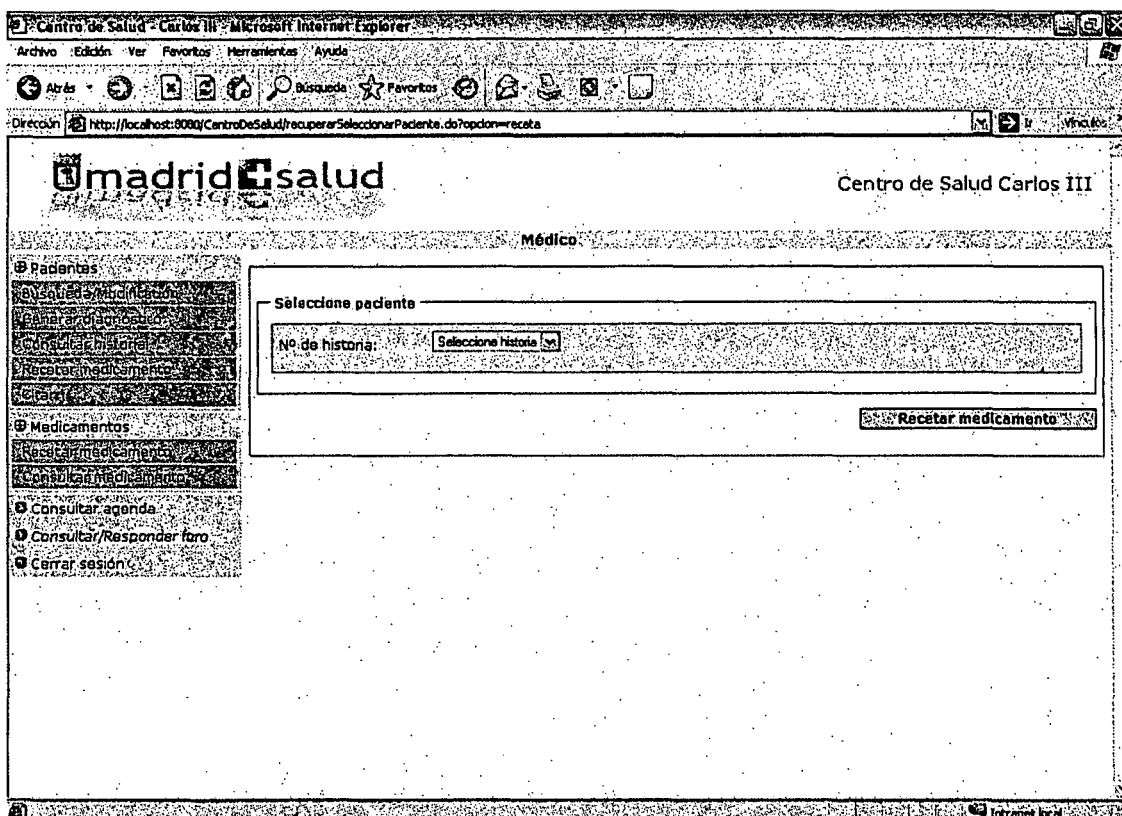


- Figura 201: Consultar diagnóstico -

5.2.5.4.- RECETAR MEDICAMENTO

Los médicos además de poder recetar un medicamento después de generar un diagnóstico ó seguimiento de Sintrom, pueden hacerlo sin necesidad de ello. Por lo general suele hacerse cuando un paciente viene solo a recoger una receta rutinaria.

Tras pulsar sobre la opción “Recetar medicamento” (Pacientes ó Medicamentos), el médico deberá seleccionar el paciente al que va a extenderle la receta, a través del combo en el cual, solo aparecerán los pacientes que el médico tiene asignados.



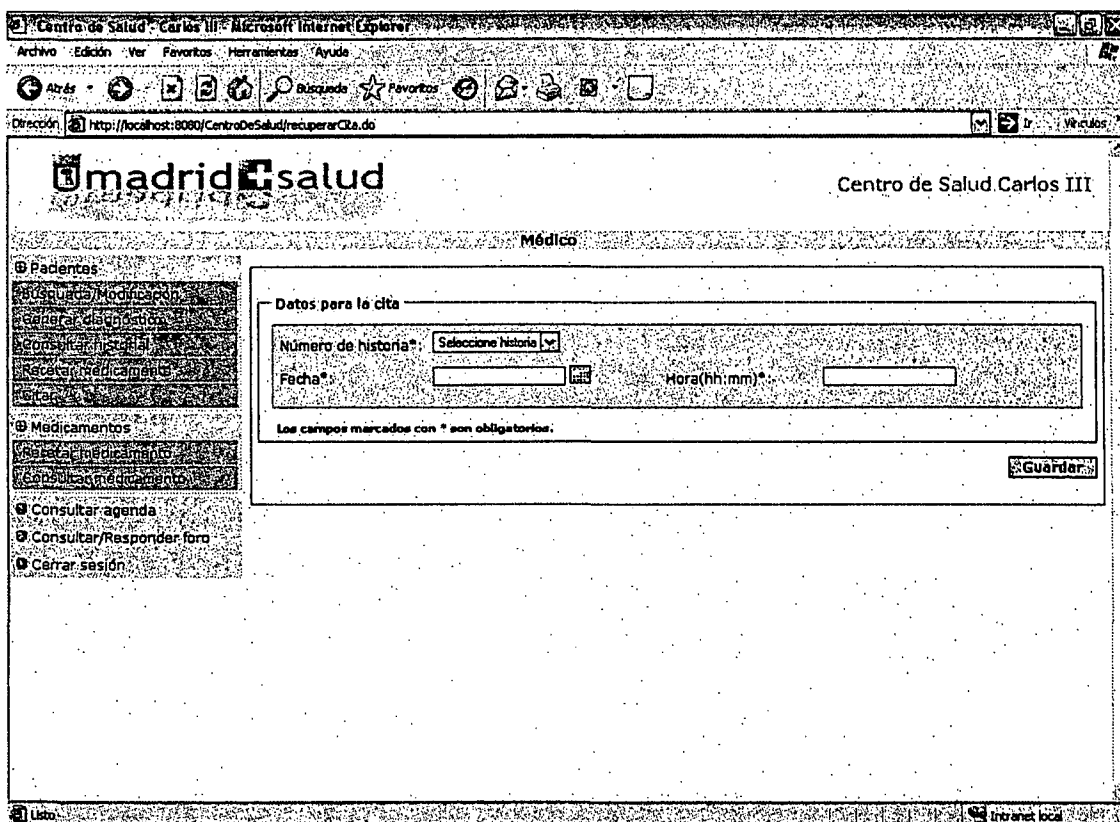
- Figura 202: Recetar medicamento -

Tras seleccionar el paciente y pulsar sobre el botón “Recetar medicamento”, el funcionamiento será el mismo que en el apartado 5.2.5.2.

5.2.5.5.- CITAR PACIENTE

Cualquier médico puede citar a cualquiera de los pacientes que tiene asignados, si necesidad de realizar primero un diagnóstico o seguimiento.

Tras pulsar sobre la opción “Citar”, aparecerá la siguiente pantalla.



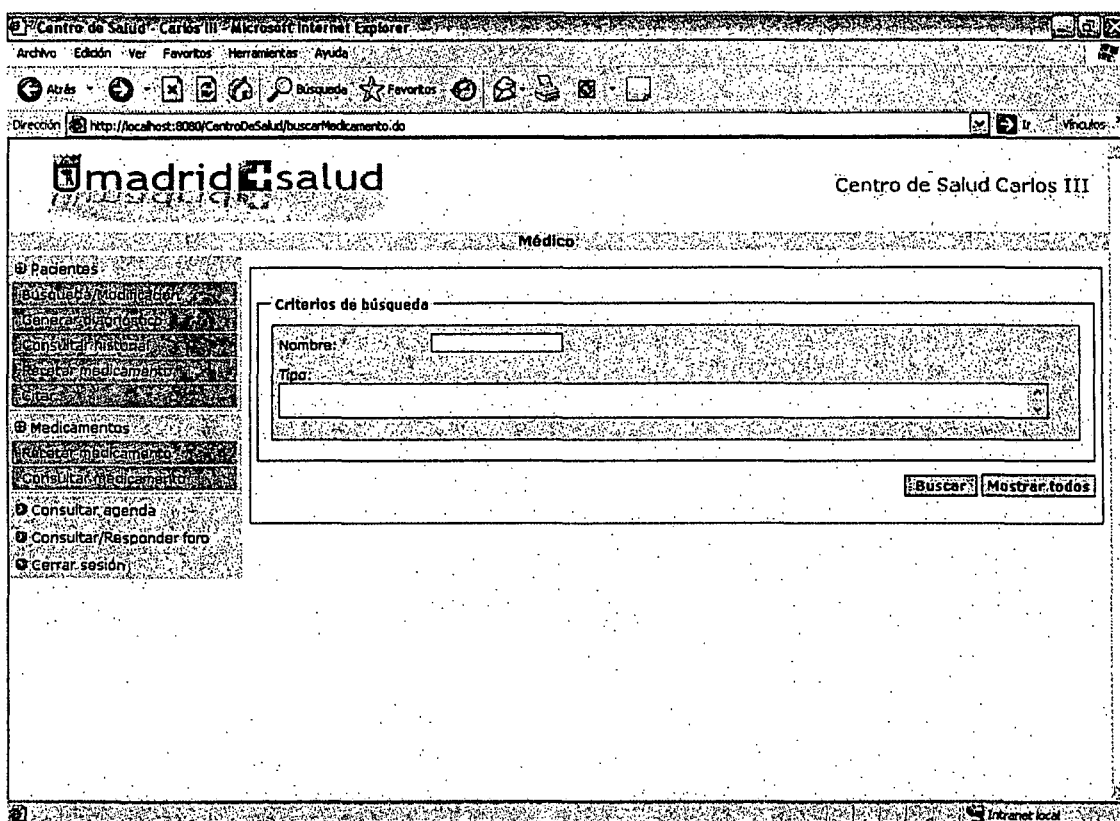
The screenshot shows a web browser window with the address bar displaying 'http://localhost:8080/CentroDeSalud/recuperarCita.do'. The page title is 'Centro de Salud Carlos III'. The main content area is titled 'Médico' and contains a form for scheduling an appointment. The form has a section 'Datos para la cita' with fields for 'Número de historia*' (with a dropdown menu 'Seleccione historia'), 'Fecha*' (with a date picker), and 'Hora(hh:mm)*' (with a time picker). A note below the fields states 'Los campos marcados con * son obligatorios.' There is a 'Guardar' button at the bottom right of the form. On the left side, there is a sidebar menu with options: 'Pacientes', 'Búsqueda/Modificación', 'Consulta de gestión', 'Consultar historia', 'Recetar medicamento', 'Historial de prescripciones', 'Medicamentos', 'Recetar medicamento', 'Consultar agenda', 'Consultar/Responder foro', and 'Cerrar sesión'.

- Figura 203: Médico cita a paciente -

El funcionamiento es el mismo que en el apartado 5.2.4.7.

5.2.5.6.- CONSULTAR MEDICAMENTO

Cualquier médico puede consultar los medicamentos de la base de datos. Para ello deberá pulsar en la opción del menú "Consultar medicamento", apareciéndole por tanto la siguiente pantalla.



- Figura 204: Consultar medicamento -

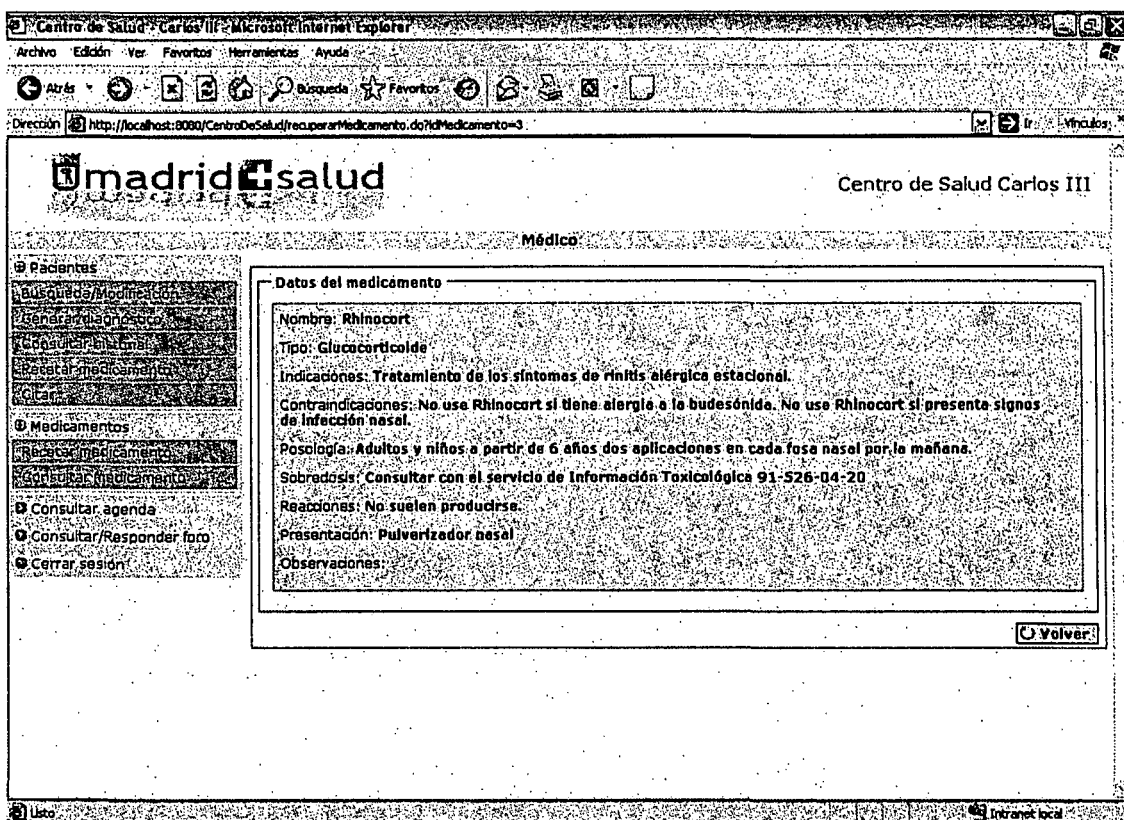
En esta pantalla se presentan dos opciones:

- Buscar un medicamento introduciendo algún/os de los criterios de búsqueda (nombre, tipo) y pulsar buscar, de forma que se muestren todos los medicamentos que coincidan con los criterios establecidos.
- Pulsar sobre el botón "Mostrar todos" para que aparezcan todos los medicamentos de la base de datos.

En ambas opciones, si no existen resultados que mostrar, aparecerá un mensaje de aviso.

Para ver imágenes visualizar apartado 5.2.4.9.

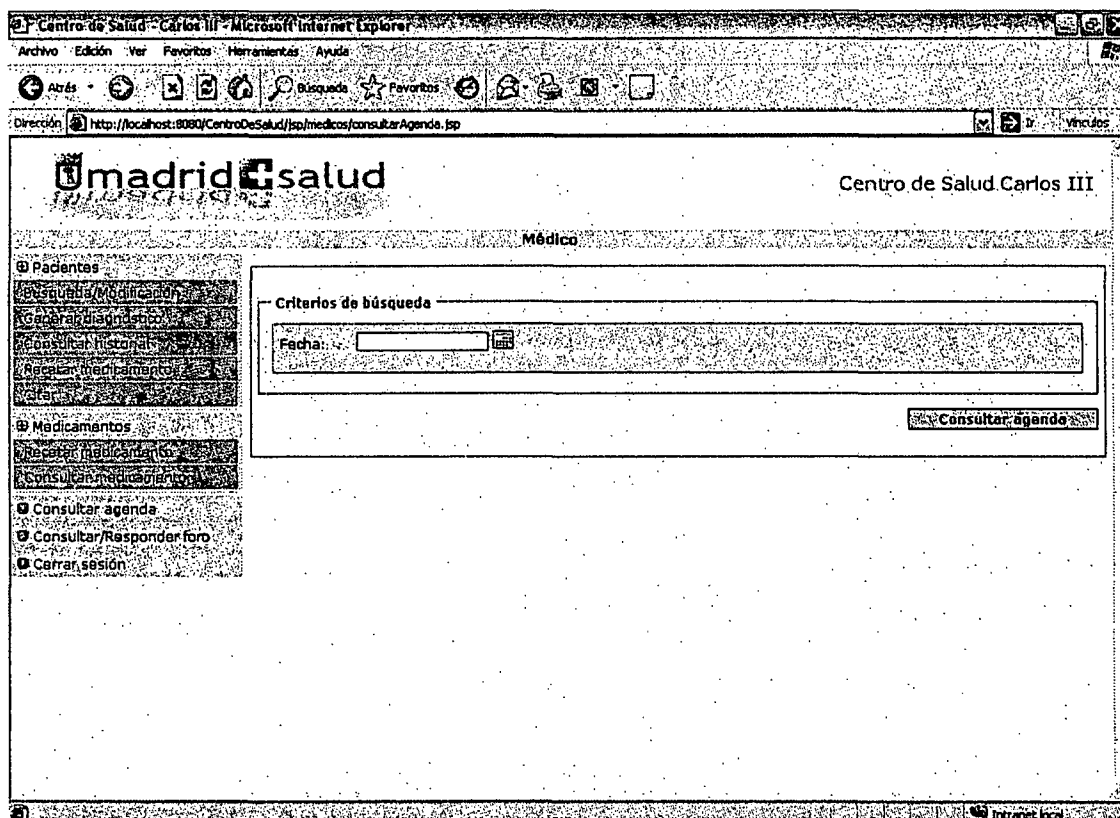
Si tras la búsqueda el médico desea ver los datos de un medicamento, solo deberá pulsar sobre su nombre, apareciendo así una pantalla con sus datos. Se da la opción de pulsar sobre el botón "Volver" para regresar a la pantalla anterior.



- Figura 205: Visualizar medicamento -

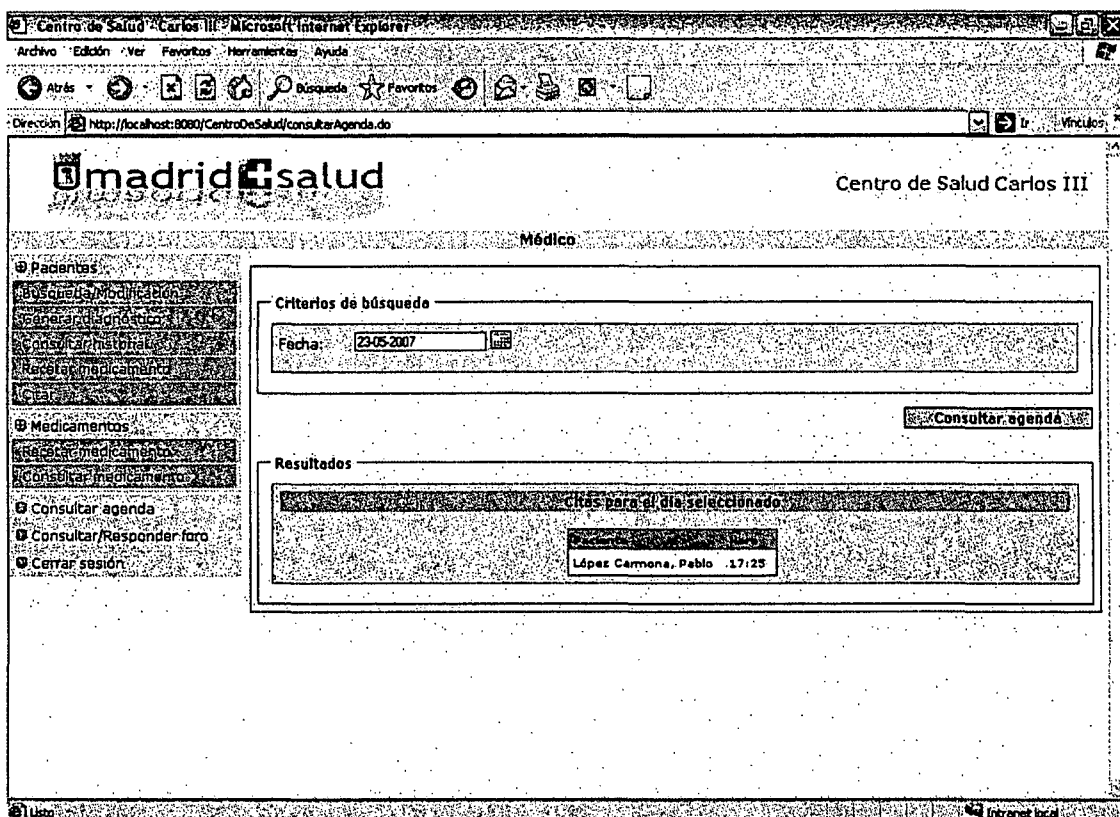
5.2.5.7.- CONSULTAR AGENDA

Los médicos pueden visualizar las citas que tienen para un día concreto, para ello deben pulsar sobre la opción "Consultar agenda" apareciendo la siguiente pantalla.



- Figura 206: Consultar agenda -

Una vez que el médico introduzca la fecha en la que desea ver sus citas y pulse el botón "Consultar agenda", aparecerá un listado con todas las citas que tiene para ese día, ó en caso de que no tenga ninguna, un mensaje de aviso.

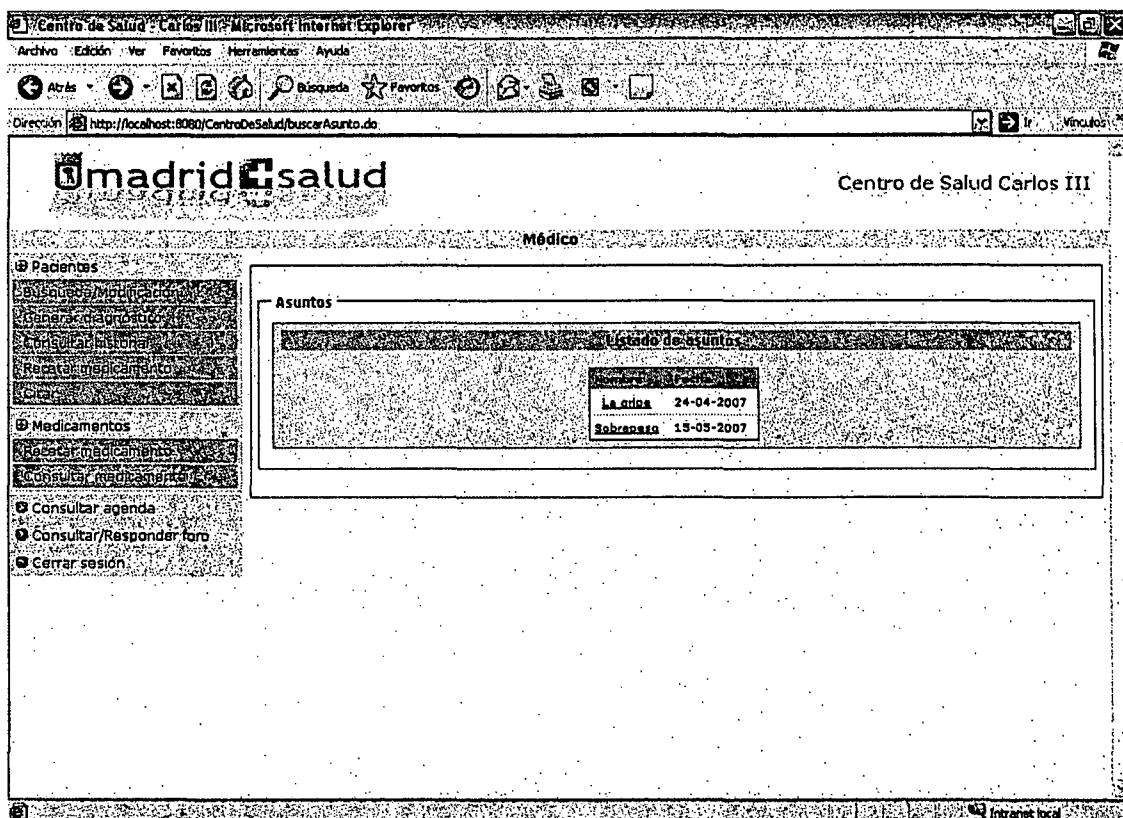


- Figura 207: Visualizar citas -

5.2.5.8.- CONSULTAR FORO Y/O RESPONDER PREGUNTA

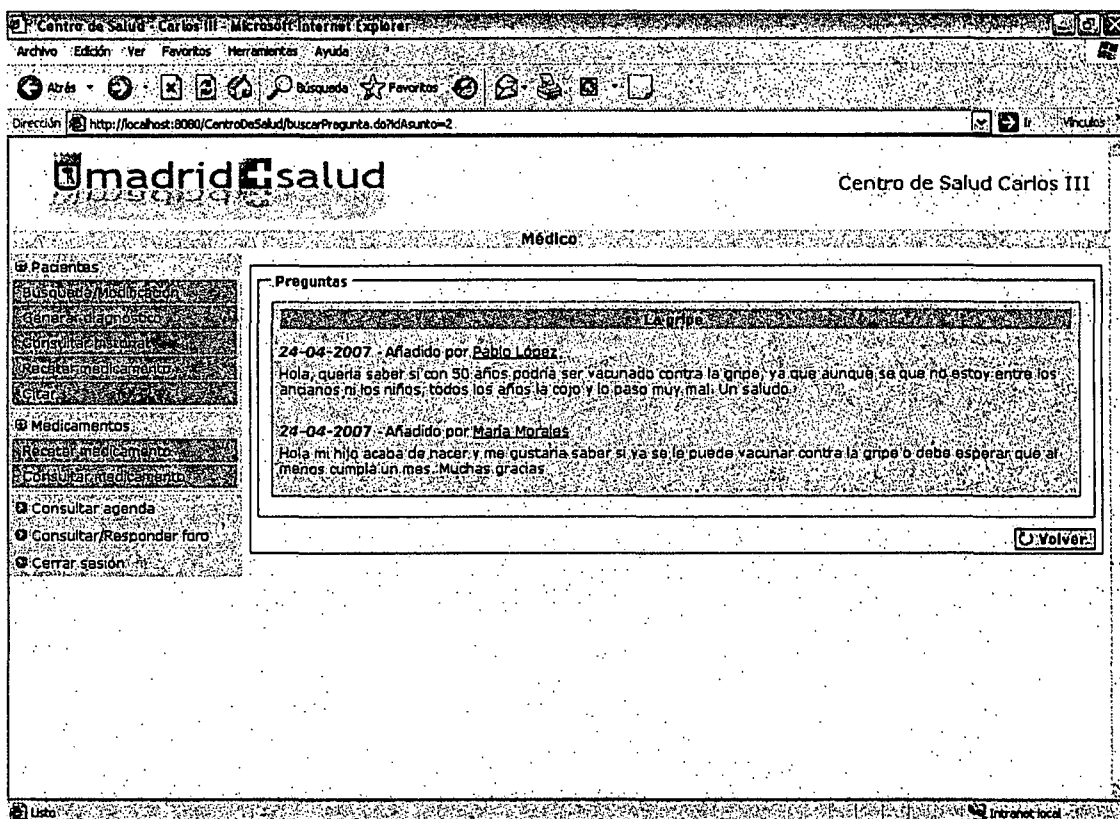
Los médicos al igual que los usuarios de la aplicación, podrán consultar el foro para ver las preguntas y respuestas de cada asunto. Además tienen opción a responder a las preguntas planteadas por los usuarios.

Tras pulsar sobre la opción del menú “Consultar/Responder foro”, aparecerá una pantalla con todos los asuntos del foro.



- Figura 208: Listado de asuntos -

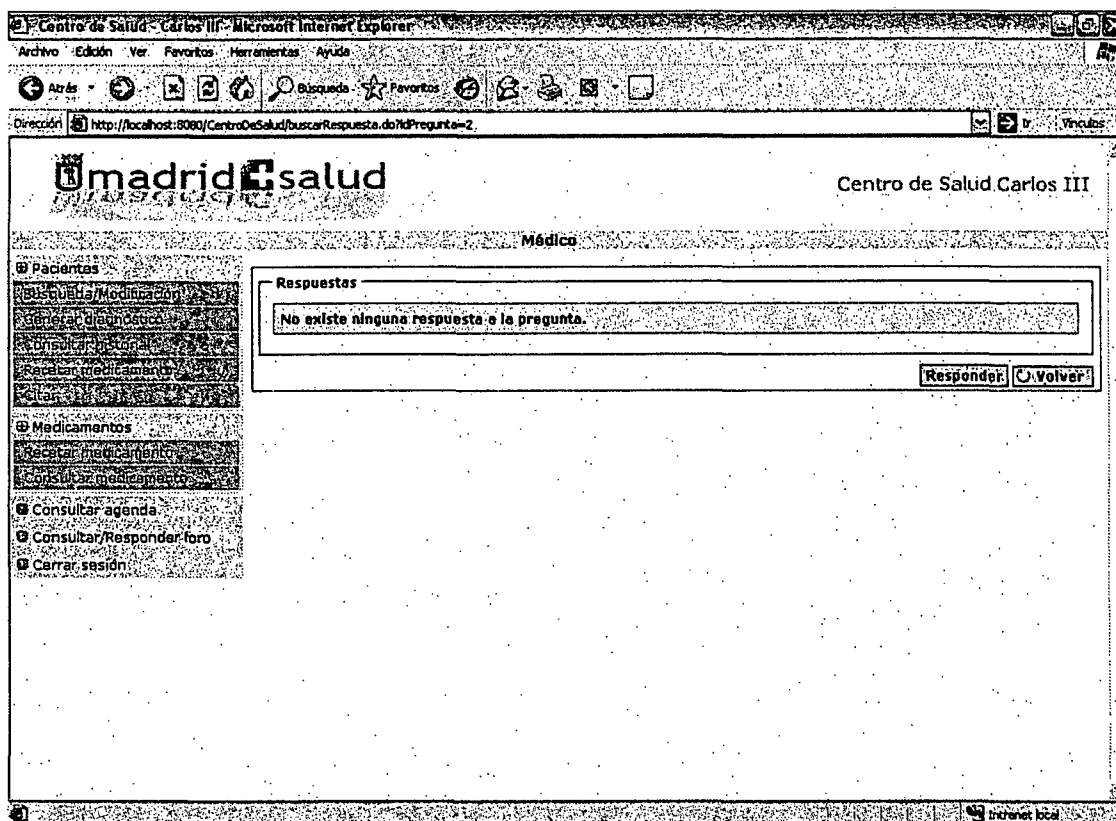
Si el médico pulsa sobre el nombre de algún asunto, le aparecerá una pantalla con todas las preguntas pertenecientes a esos asuntos, realizadas por los usuarios, ó un mensaje de aviso, en caso de no existir ninguna pregunta.



- Figura 209: Listado de preguntas -

En esta pantalla, además existen dos opciones:

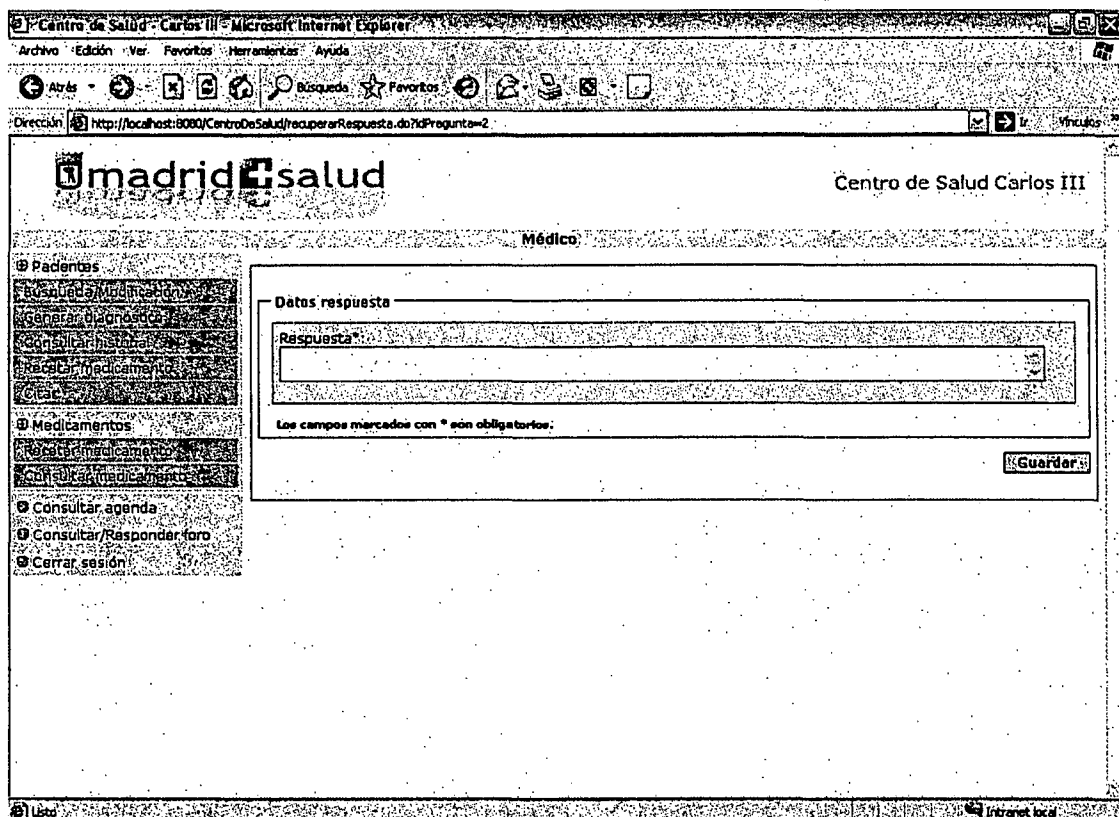
- Volver: Tras pulsar en esta opción, se redirigirá a la pantalla anterior, es decir, al listado de asuntos de foro.
- El médico puede pulsar sobre el nombre de la persona que ha introducido una pregunta, en caso de existir alguna, y visualizar así todas las respuestas que han dado los médicos del centro a esa pregunta, ó un mensaje de aviso, en caso de no existir ninguna respuesta.



- Figura 210: No existe ninguna respuesta -

En esta pantalla, existen otras dos opciones:

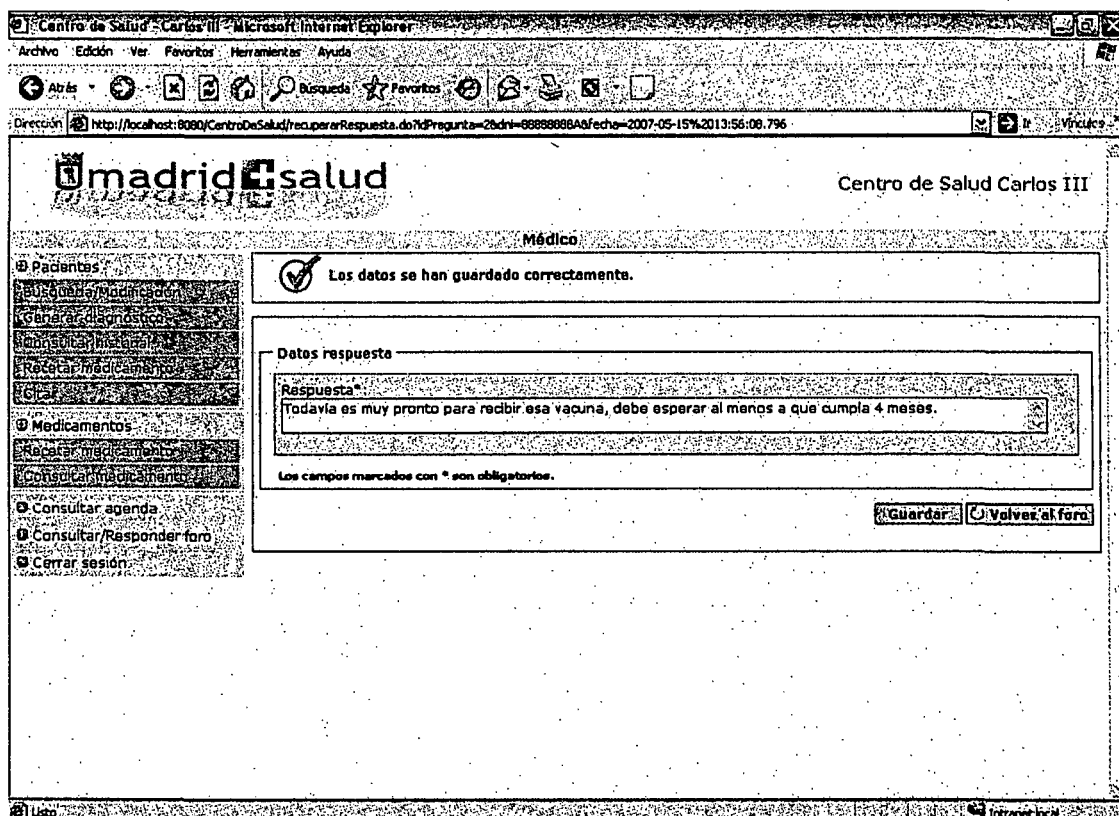
- Volver: Tras pulsar en esta opción, se redirigirá a la pantalla anterior, es decir, al listado de preguntas.
- Responder: Si el médico pulsa este botón podrá responder a la pregunta planteada, introduciendo la respuesta deseada y pulsando sobre el botón "Guardar".



- Figura 211: Responder -

En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo la respuesta introducida, el médico podrá modificarla y tras pulsar de nuevo en “Guardar” estos se actualizarán.

Pulsando sobre el botón “Volver” se redirecciona a la pantalla de respuesta, con la nueva respuesta añadida.



- Figura 212: Respuesta guardada con éxito -

5.2.6.- CERRAR SESIÓN

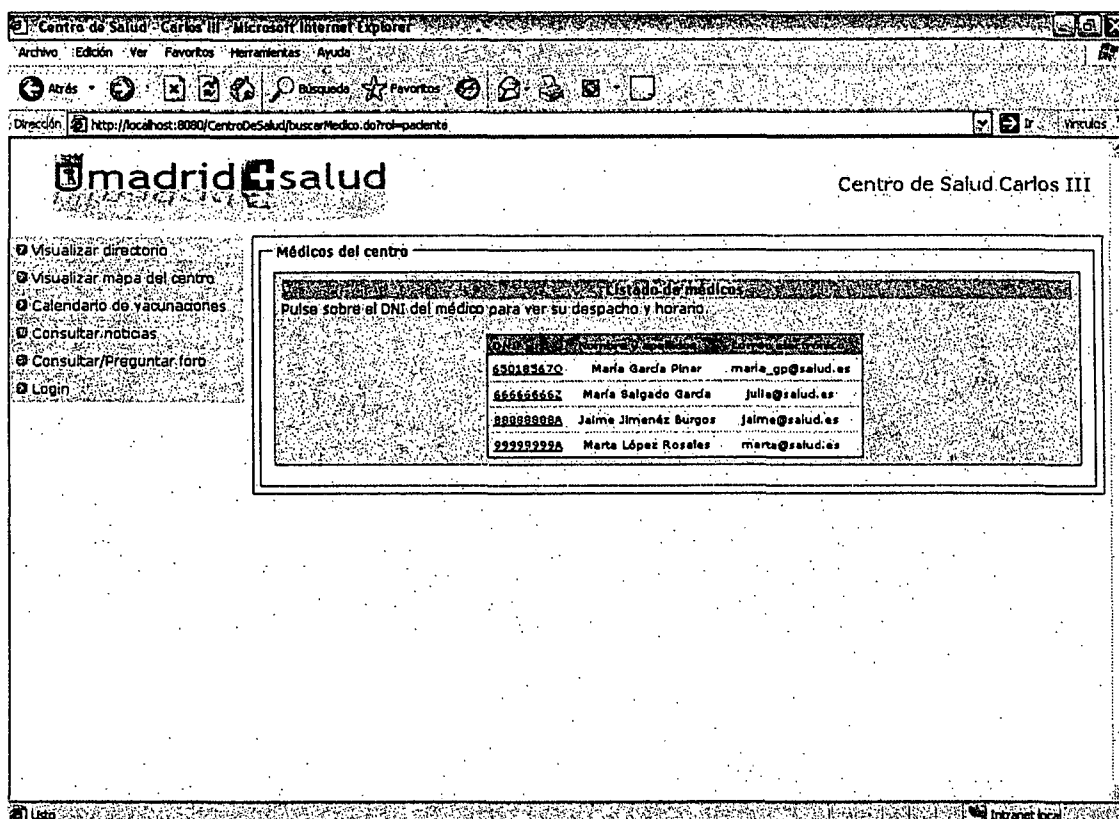
Tanto el administrador como el médico, tienen en su menú la opción de cerrar sesión. Si pulsan sobre esta opción, se cerrará su sesión de usuario y se redireccionará a la pantalla principal.

5.2.7.- OPCIONES DE CUALQUIER USUARIO NO LOGADO

5.2.7.1.- VISUALIZAR DIRECTORIO DE MÉDICOS

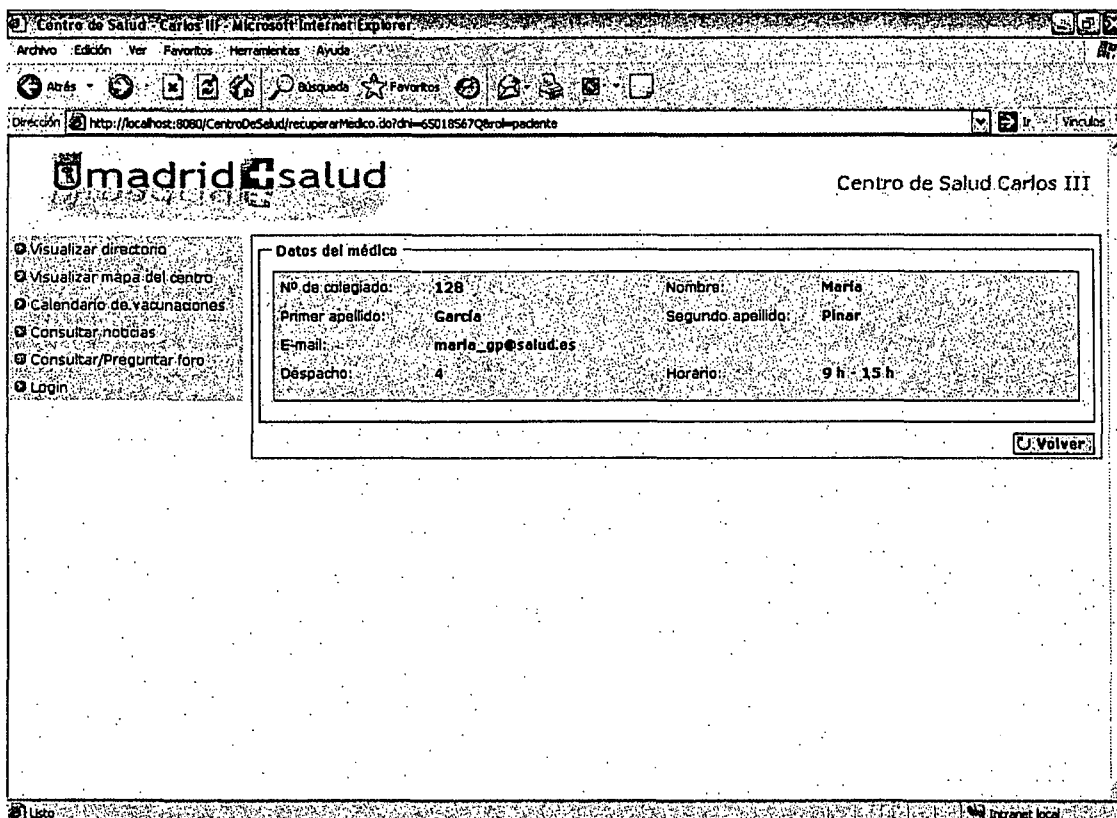
Cualquier usuario, tiene la posibilidad de ver los datos de todos los médicos del centro, para así por ejemplo saber cual es su despacho, su turno,...

Para ello deben pulsar en la opción del menú "Visualizar directorio", de manera que les aparecerá un listado con todos lo médicos.



- Figura 213: Listado de médicos del centro -

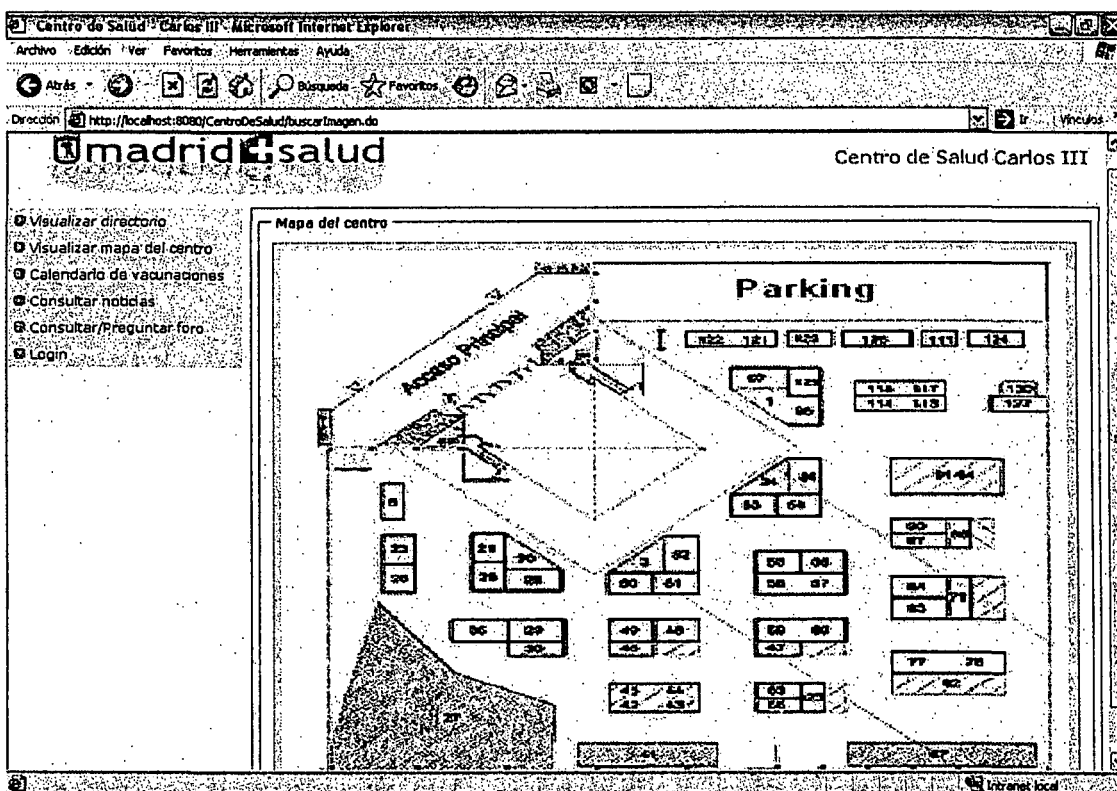
Si desean consultar algún médico en particular, deben pulsar sobre el DNI del médico, de forma que les aparezca una pantalla con todos los datos necesarios. Además en esa pantalla existe el botón “Volver” para que regresen a la pantalla anterior y puedan si lo desean, consultar más médicos.



- Figura 214: Visualizar datos del médico -

5.2.7.2.- VISUALIZAR MAPA DEL CENTRO

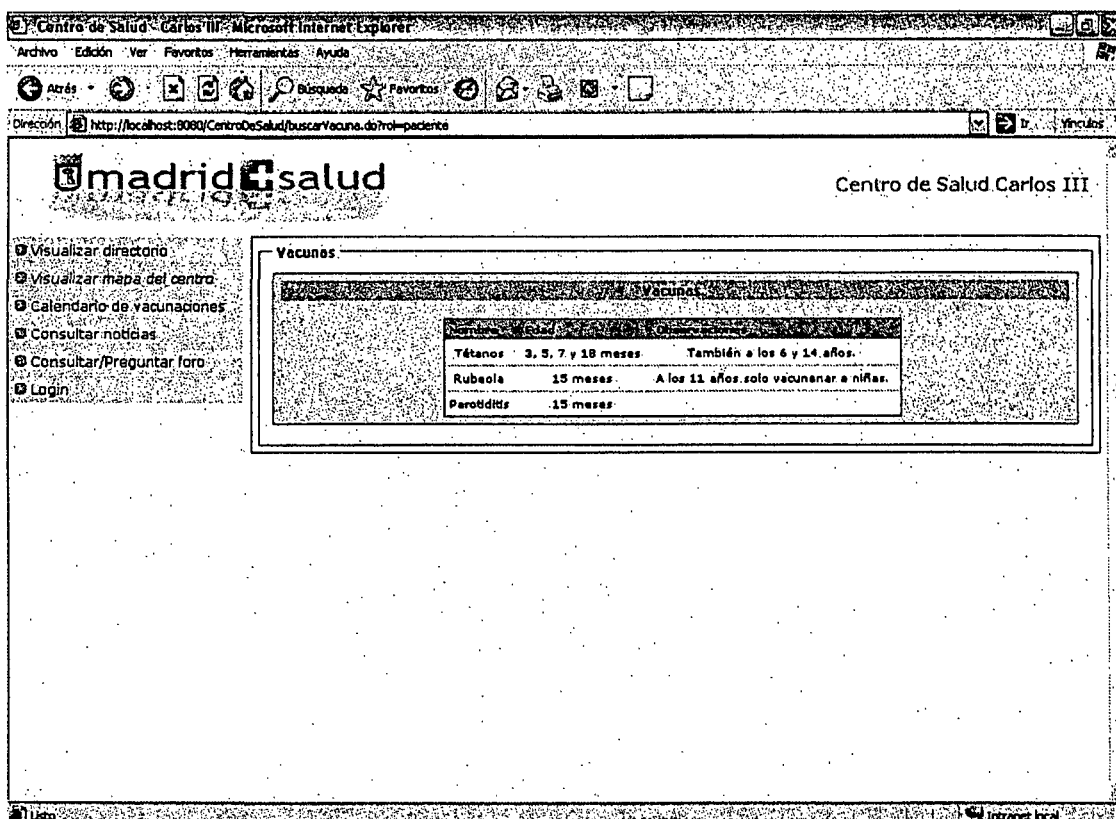
Cualquier usuario podrá ver un plano del centro si pulsa sobre la opción del menú "Visualizar mapa del centro". Esta opción será útil para que sepan la dirección del centro, y el donde se encuentra el despacho de cada médico.



- Figura 215: Visualizar mapa -

5.2.7.3.- CALENDARIO DE VACUNACIONES

Todos los usuarios de la página, podrán seleccionar la opción del menú “Calendario de vacunaciones”, para poder ver todas las vacunas que existen en ese momento y la edad a la que deben ser puestas.



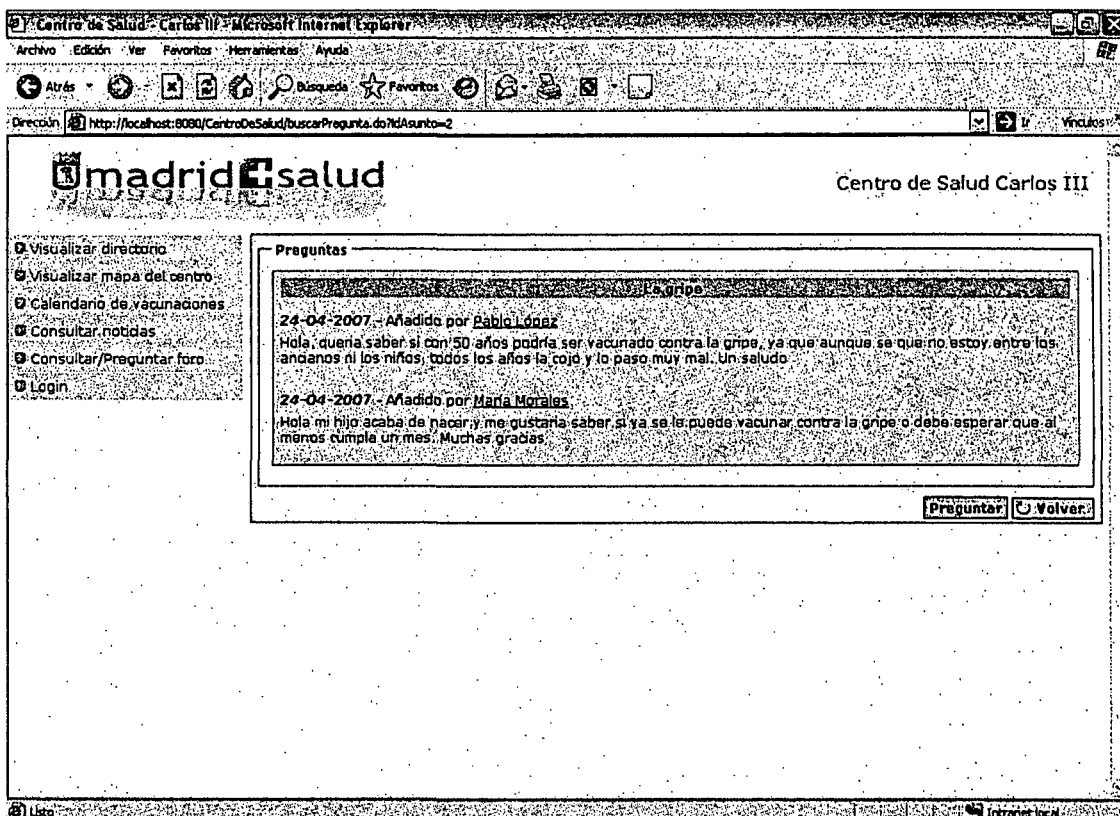
- Figura 216: Calendario de vacunaciones -

5.2.7.4.- CONSULTAR FORO Y/O FORMULAR PREGUNTA

Los usuarios de la aplicación, podrán consultar el foro para ver las preguntas y respuestas de cada asunto. Además tienen opción de realizar preguntas que posteriormente podrán ser respondidas por los médicos del centro.

Tras pulsar sobre la opción del menú “Consultar/Preguntar foro”, aparecerá una pantalla con todos los asuntos del foro.

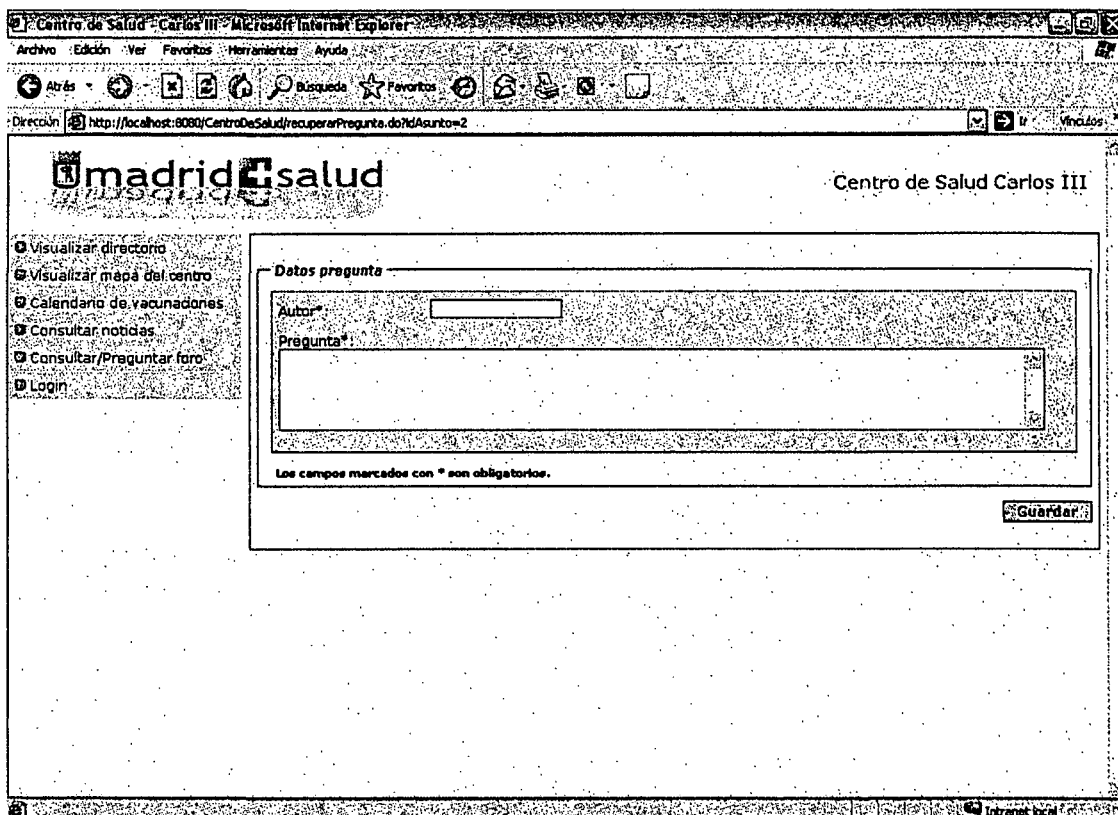
Si el usuario pulsa sobre el nombre de algún asunto, le aparecerá una pantalla con todas las preguntas pertenecientes a esos asuntos, realizadas por los usuarios, ó un mensaje de aviso, en caso de no existir ninguna pregunta.



- Figura 217: Listado de preguntas -

En esta pantalla, además existen tres opciones:

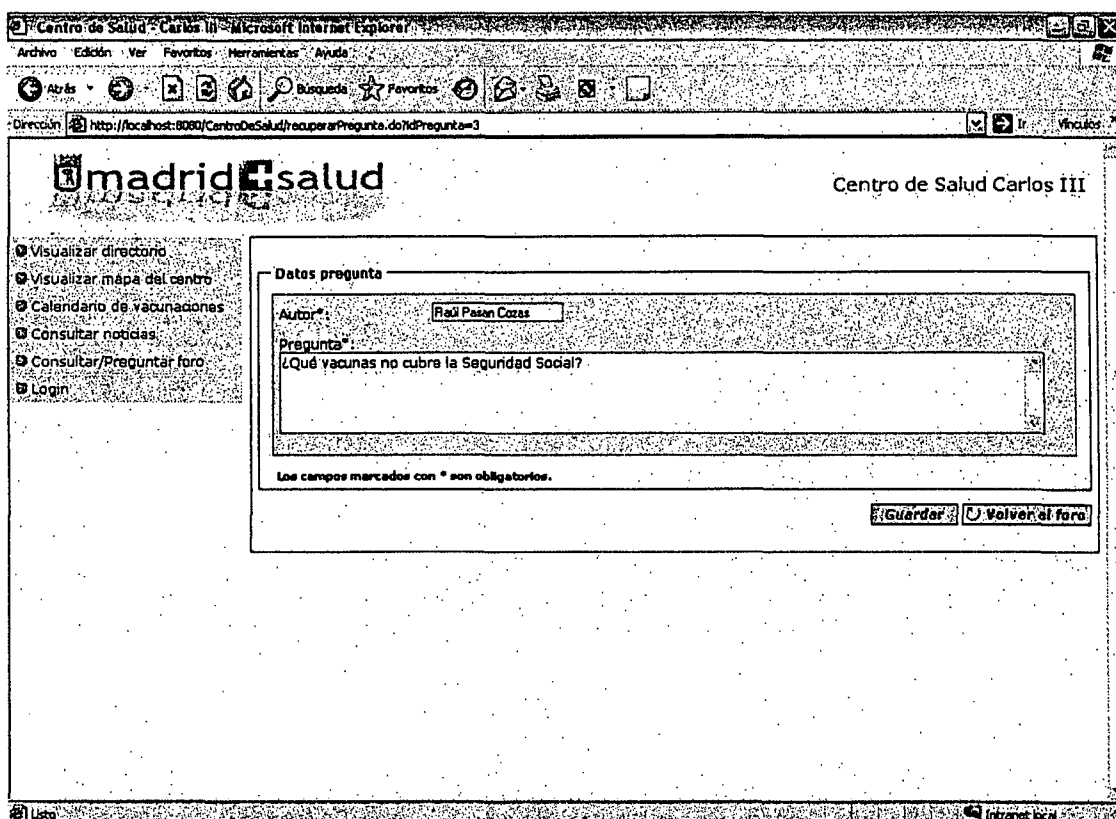
- Volver: Tras pulsar en esta opción, se redirigirá a la pantalla anterior, es decir, al listado de asuntos de foro.
- Preguntar: Si un usuario pulsa este botón podrá realizar una pregunta, introduciendo los datos necesarios y pulsando sobre el botón "Guardar".



- Figura 218: Preguntar -

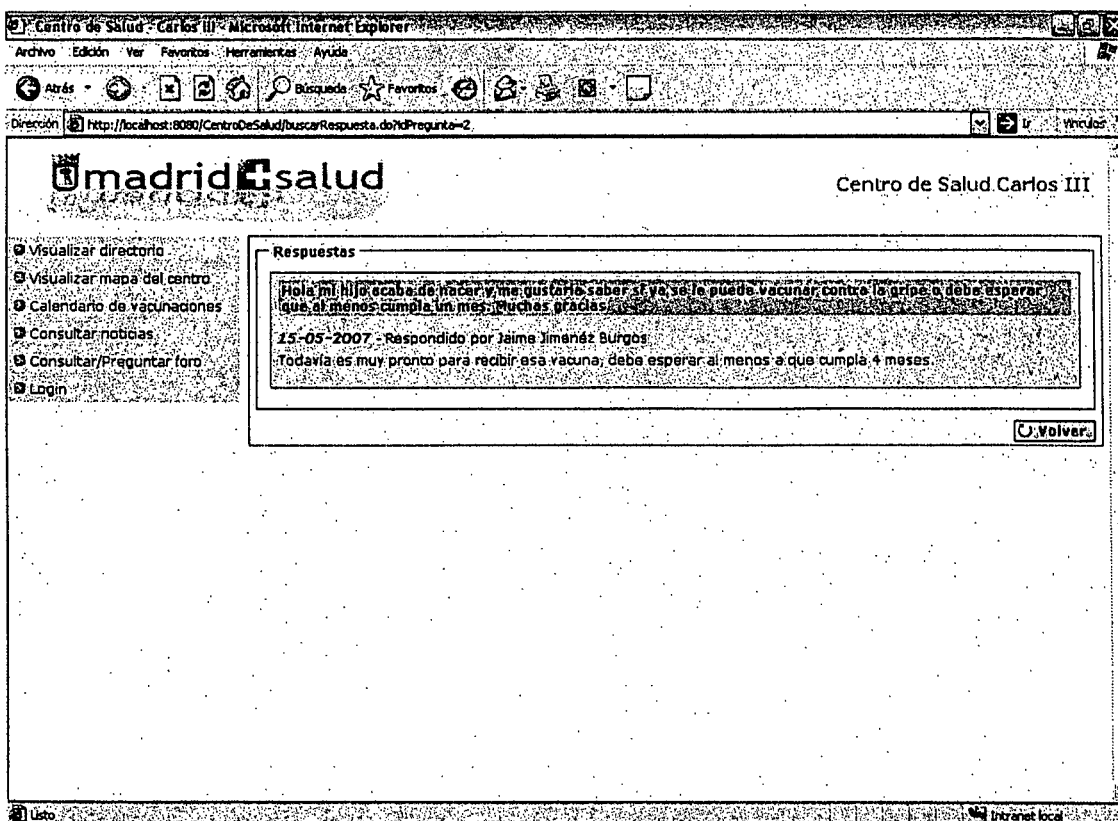
En caso de producirse algún error se avisará dando la oportunidad de solventarlo. Si todo ha ido correctamente, se mostrará un mensaje y se permitirá visualizar los datos introducidos para verificar que están correctos. En caso de encontrar algún fallo la pregunta introducida, el usuario podrá modificarla y tras pulsar de nuevo en “Guardar” estos se actualizarán.

Pulsando sobre el botón “Volver” se redireccionará a la pantalla de preguntas, con la nueva pregunta añadida.



- Figura 219: Pregunta guardada con éxito -

- El usuario puede pulsar sobre el nombre de la persona que ha introducido una pregunta, en caso de existir alguna, y visualizar así todas las respuestas que han dado los médicos del centro a esa pregunta, ó un mensaje de aviso, en caso de no existir ninguna respuesta. En esa pantalla el usuario puede pulsar sobre el botón “Volver” para regresar al listado de preguntas.



- Figura 220: Listado de respuestas -



6.- ACCESIBILIDAD

6.1.- INTRODUCCIÓN

El Consorcio para la World Wide Web (W3C) fue fundado en octubre de 1994 para conducir a la World Wide Web (WWW) a su máximo potencial desarrollando protocolos de uso común que promocionaran su evolución y aseguraran su interoperabilidad.

El compromiso del W3C de encaminar la Web a su máximo potencial incluye promover un alto grado de accesibilidad para las personas con discapacidad. El grupo de trabajo permanente **Web Accessibility Initiative** (WAI, Iniciativa para la Accesibilidad de la Red) en coordinación con organizaciones de todo el mundo, persigue la accesibilidad de la Web a través de cinco áreas de trabajo principales: tecnología, directrices, herramientas, formación y difusión, e investigación y desarrollo. El WAI está dirigido por Judy Brewer, Directora del Programa Internacional, y por Daniel Dardailler como Director de Proyectos.

La accesibilidad de un Sitio Web se refiere a su capacidad para presentar contenidos a personas que cuentan con discapacidades físicas, que les impiden usar la información disponible de una manera tradicional y por ello emplean ayudas técnicas.

Para verificar la accesibilidad de una Web, la WAI ha creado una serie de pautas de accesibilidad, que según su grado de prioridad darán la página un mayor o menor nivel de accesibilidad.

Las prioridades establecidas son las siguientes:

- **Prioridad 1 (A):** Un desarrollador tiene que satisfacer este punto de verificación, ya que sino, uno o más grupos de usuarios encontrarán imposible acceder a la información del documento. Satisfacer este punto de verificación es un requerimiento básico para que algunos grupos puedan usar los documentos Web.
- **Prioridad 2 (AA):** Un desarrollador debe satisfacer este punto de verificación, de otra forma, uno o más grupos encontrarán dificultades en el acceso a la información del documento. Satisfacer este punto de verificación eliminará importantes barreras de acceso a los documentos Web.
- **Prioridad 3 (AAA):** Un desarrollador puede satisfacer este punto de verificación, en caso contrario, uno o más grupos de usuarios encontrarán alguna dificultad para acceder a la información del documento. Satisfacer este punto de verificación mejorará la accesibilidad de los documentos Web.

En la realización de este proyecto, se ha buscado cumplir al menos la doble A (prioridad 2) en todas las páginas de acceso público. Mientras que en el resto de página se ha pretendido cumplir únicamente A (prioridad 1). A pesar de ello, muchas páginas cumplen la triple A (prioridad 3).



6.2.- PAUTAS DE ACCESIBILIDAD

En este apartado, se realiza un breve resumen de las diferentes pautas de accesibilidad, explicando la manera en la que se han logrado cumplir, en caso, de ser necesario.

6.2.1.- PAUTA 1

La primera pauta consiste en proporcionar alternativas equivalentes para el contenido visual y auditivo.

Algunas personas no pueden utilizar imágenes, películas, sonidos..., directamente, en cambio, pueden hacer uso de páginas que incluyen la información equivalente a los contenidos visuales y auditivos. Para ello, es necesario que todas las imágenes contengan una descripción asociada a ellas, mediante la cual a través de un sintetizador de voz pueda ser interpretado, ó impreso en braille mediante un dispositivo braille.

El sintetizador de voz es esencial para personas ciegas y para las que tienen dificultades de lectura que a menudo acompañan a discapacidades cognitivas, de aprendizaje o sordera. El braille es esencial para personas sordo-ciegas, además de para muchos individuos que solamente son ciegos.

Además es importante también representar imágenes para beneficiar en este caso a personas analfabetas o con dificultad para la lectura.

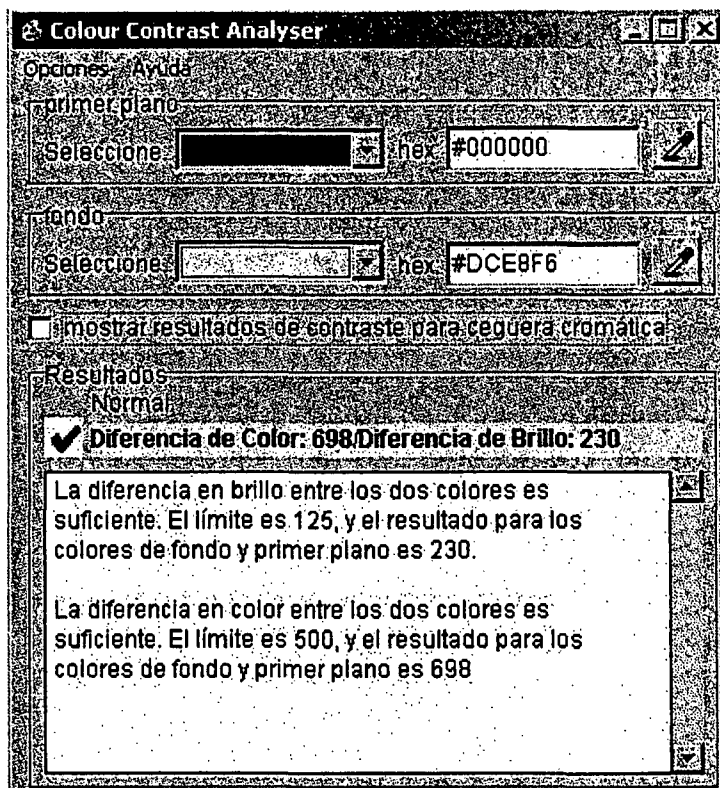
6.2.2.- PAUTA 2

La segunda pauta consiste en asegurarse que los textos y gráficos son comprensibles cuando se ven sin color, ya que algunas personas no pueden diferenciar bien todos los colores ó confunden unos con otros como son los daltónicos, además muchos usuarios pueden no tener pantallas a color.

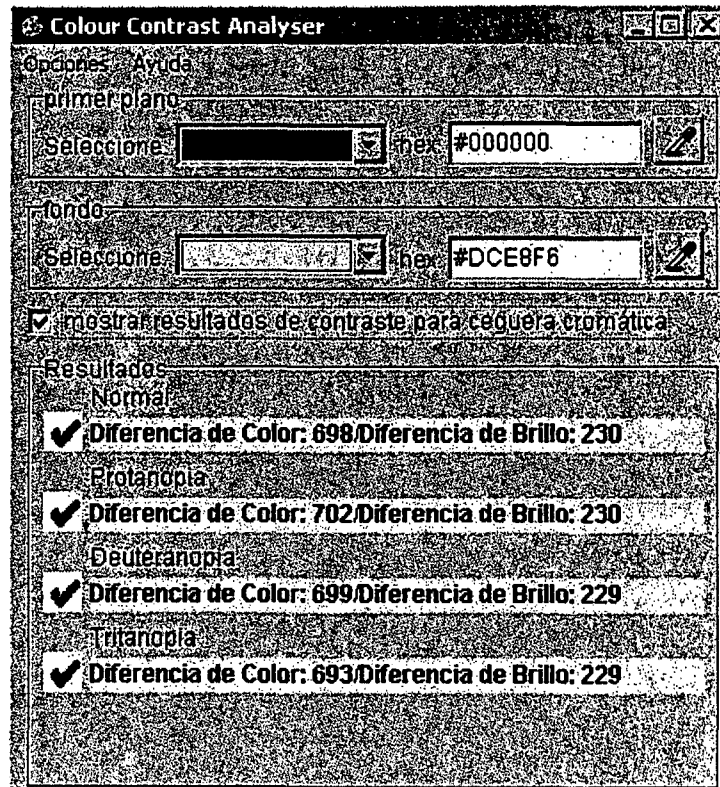
Cuando los colores de primer plano y de fondo tienen un tono similar, pueden no proporcionar suficiente contraste en las pantallas monocromáticas, así como a las personas con diferentes tipos de deficiencias de percepción de los colores.

Para poder cumplir esta pauta, ha sido necesario utilizar un analizador de contrastes antes de utilizar un color en cada página.

El programa recibe el nombre de **Analizador de Contraste de Color 1.0**, ha sido desarrollado por Jun en colaboración con Steve Faulkner.



- Figura 221: Analizador de contrastes 1 -



- Figura 222: Analizador de contrastes 2 -



6.2.3.- PAUTA 3

La tercera pauta trata de marcar los documentos con los elementos estructurales apropiados y controlar la presentación mediante hojas de estilos.

Para cumplir esta pauta ha sido necesario utilizar los siguientes elementos:

- Creación de documentos validados por las gramáticas formales publicadas, es decir, utilizando las DTD de referencia.
- Uso de hojas de estilo CSS para controlar la maquetación y la presentación.
- Utilización de unidades relativas en lugar de absolutas para especificar los valores en los atributos y propiedades de las hojas de estilo, es decir, usar “em” en vez de medidas como puntos o centímetros.
- Uso de elementos de encabezado como “h1”, “h2”, “fieldset”... para transmitir una estructura lógica.
- Utilización de elementos de tipo lista, para ordenar sus elementos. (“li”, “ul”, “th” y “td”).

6.2.4.- PAUTA 4

La cuarta pauta habla sobre la identificación del idioma usado mediante el uso de marcadores que faciliten la pronunciación o interpretación de texto abreviado o extranjero.

En este proyecto, esto se ha conseguido a través del uso del atributo “xml:lang” y de “title” añadidos a algunos elementos como las imágenes. De esta forma, los sintetizadores de voz y los dispositivos braille pueden cambiar automáticamente al nuevo idioma, consiguiendo que el documento sea accesible a usuarios multilingües.

6.2.5.- PAUTA 5

La quinta pauta trata sobre la necesidad de crear las tablas utilizando los marcadores necesarios para que puedan ser interpretadas por el navegador.

Es necesario utilizar los elementos “th” y “td”, además de añadir resúmenes a dichas tablas para especificar su contenido.



6.2.6.- PAUTA 6

La sexta pauta indica que debemos asegurarnos de que las páginas funcionan cuando están desactivadas algunas tecnologías modernas.

6.2.7.- PAUTA 7

La séptima pauta nos indica que debemos asegurarnos de que los objetos o páginas que se mueven, parpadean, se desplazan o se actualizan automáticamente, puedan ser detenidos o parados.

Esto es necesario porque algunas personas con discapacidades cognitivas o visuales son incapaces de leer textos que se mueven con la suficiente rapidez. El movimiento puede también distraer de tal manera que el resto de la página se vuelve ilegible para las personas con discapacidades cognitivas.

6.2.8.- PAUTA 8

La octava pauta implica que la interfaz de usuario siga todos los principios de un diseño accesible, es decir, que cualquier funcionalidad sea independiente del modo de acceso a dicha página.

6.2.9.- PAUTA 9

La novena pauta trata sobre el diseño de las páginas para la independencia del dispositivo, utilizando así características que permitan la activación de los elementos de una página a través de dispositivos de entrada.

Cualquier usuario debe poder interactuar con la aplicación a través de un dispositivo de entrada ó salida (ratón, teclado, voz...).

Generalmente, las páginas que permiten la interacción a través del teclado, como son las de este proyecto, son también accesibles a través de una entrada de voz o una serie de comandos.

6.2.10.- PAUTA 10

La décima pauta indica el uso de soluciones de accesibilidad provisionales, útiles para el acceso por ejemplo con navegadores antiguos.



6.2.11.- PAUTA 11

La undécima pauta se refiere al uso de las tecnologías y pauta de la W3C.

Para este proyecto se han evitado el uso de elementos desaconsejados por la W3C, incluyéndolos en hojas de estilo en cascada (CSS).

6.2.12.- PAUTA 12

La duodécima pauta indica el uso de información de contexto y orientativa para ayudar a los usuarios a entender las páginas y los elementos complejos.

Para lograrla en este proyecto se han utilizado diversos “*title*” en las páginas, además de añadir en todas las páginas “*fieldset*” y “*legend*” consiguiendo así dividir los bloques largos de información en grupos mas manejables. Por otro lado, las cajas de texto contienen todas sus etiquetas de control correspondiente.

6.2.13.- PAUTA 13

Esta pauta pretende que se proporcionen diversos mecanismos de navegación claros y coherentes, que hagan al usuario su navegación por una página mucho más fácil y amigable.

En este proyecto para cumplir dicha pauta se han añadido diversos botones de colores llamativos, hipervínculos, un menú para navegar, leyendas, mensajes de ayuda y otros muchos elementos que hacen al usuario más fácil la navegación por la aplicación.

6.2.14.- PAUTA 14

La última pauta indica la necesidad de que el lenguaje sea claro y comprensible por todos los usuarios de las páginas, además de tener una organización coherente y simple.

Esto se consigue creando estructuras comunes para las páginas que mantengan ciertos elementos en las mismas posiciones para no confundir a los usuarios.

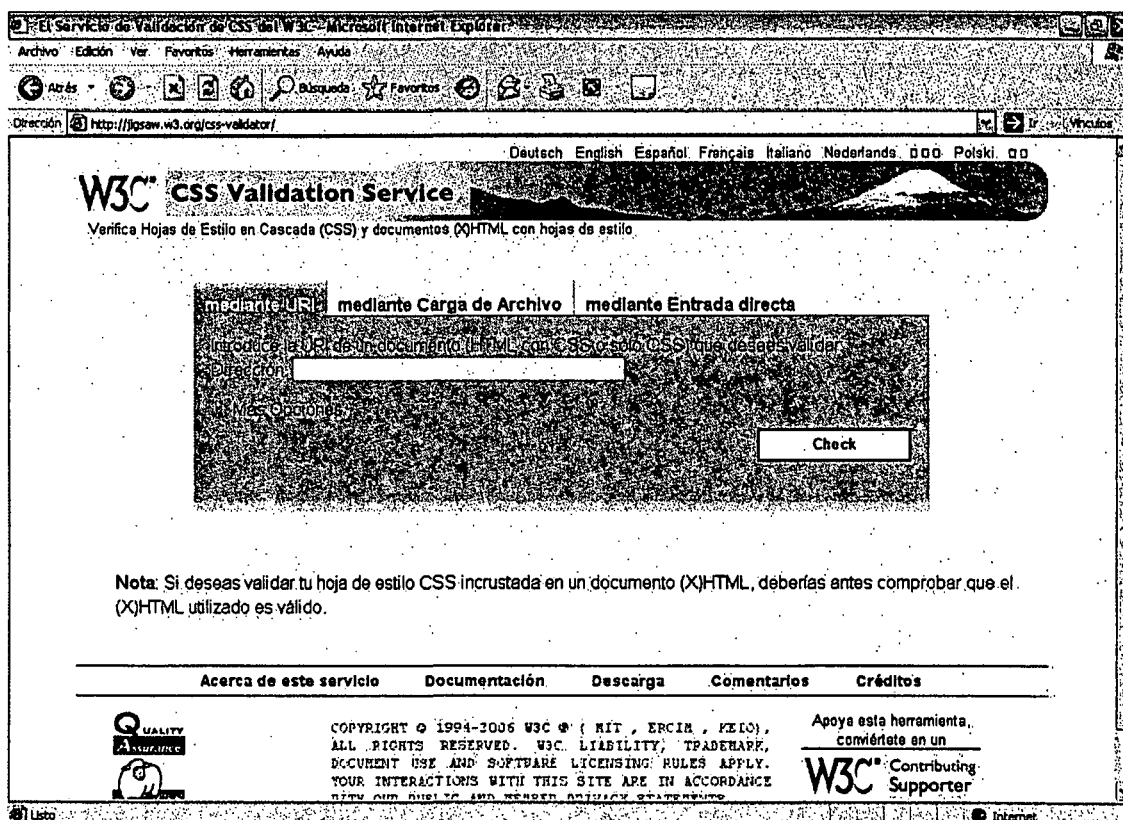
6.3.- ANÁLISIS DE ACCESIBILIDAD

Para realizar el análisis de accesibilidad al proyecto, se ha utilizado un validador de hojas de estilo en cascada y otro de accesibilidad, ambos pertenecientes a la WAI.

Los análisis de accesibilidad se han realizado en la parte pública de la aplicación, es decir, donde tiene acceso cualquier usuario de Internet, ya que es en estas páginas donde mas importante es que se cumplan las reglas de accesibilidad. Además de analizar otras de las páginas de acceso privado, en este caso, no han sido todas, ya que la mayoría están basadas en la misma plantilla.

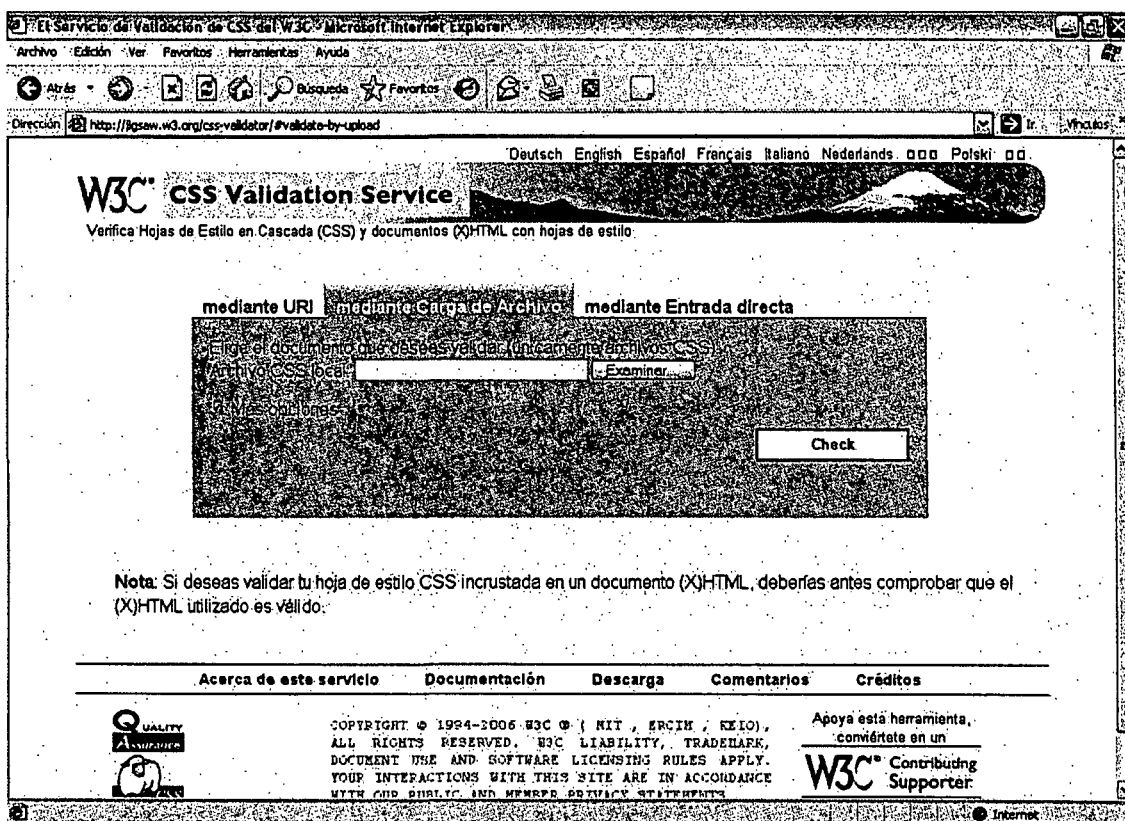
6.3.1.- VERIFICACIÓN DE HOJAS DE ESTILO EN CASCADA

Para verificar las hojas de estilo del proyecto, es necesario introducir la siguiente dirección <http://jigsaw.w3.org/css-validator/> en un navegador Web, de forma que aparezca la siguiente pantalla.



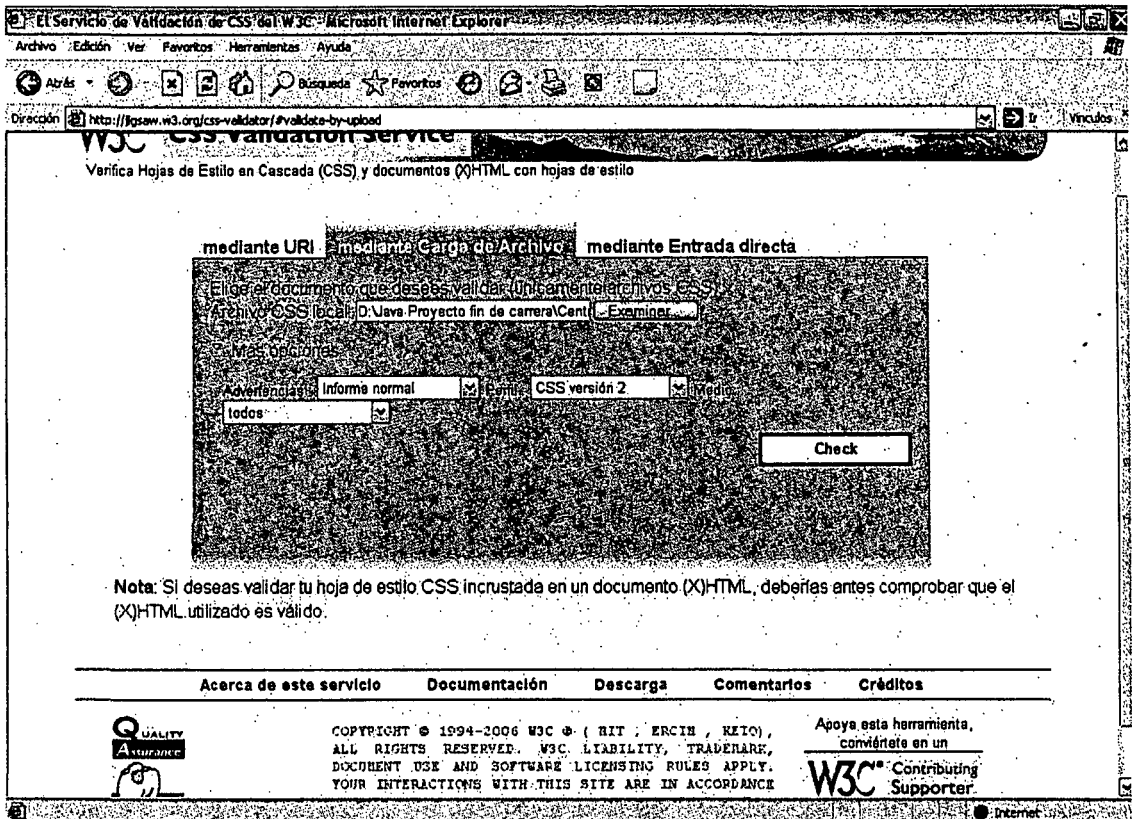
- Figura 223: Validación de CSS 1 -

Después se debe pulsar en “mediante Carga de Archivo”, de forma que podamos cargar la hoja de estilos de la aplicación para verificarla. En este caso será “*estilos.css*”.



- Figura 224: Validación de CSS 2 -

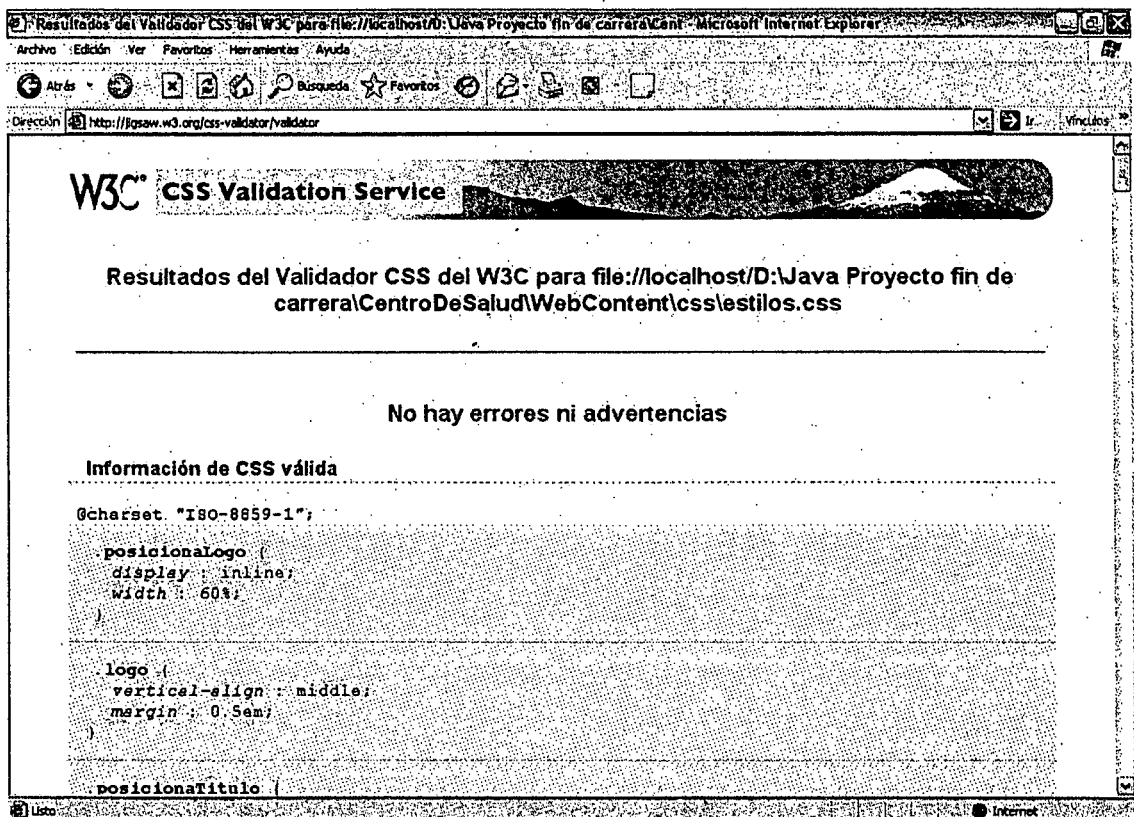
Tras cargar el archivo a analizar, debemos pulsar sobre “mas opciones” para señalar en el perfil la opción de CSS versión 2, que es el usado para desarrollar el proyecto.



- Figura 225: Validación de CSS 3 -

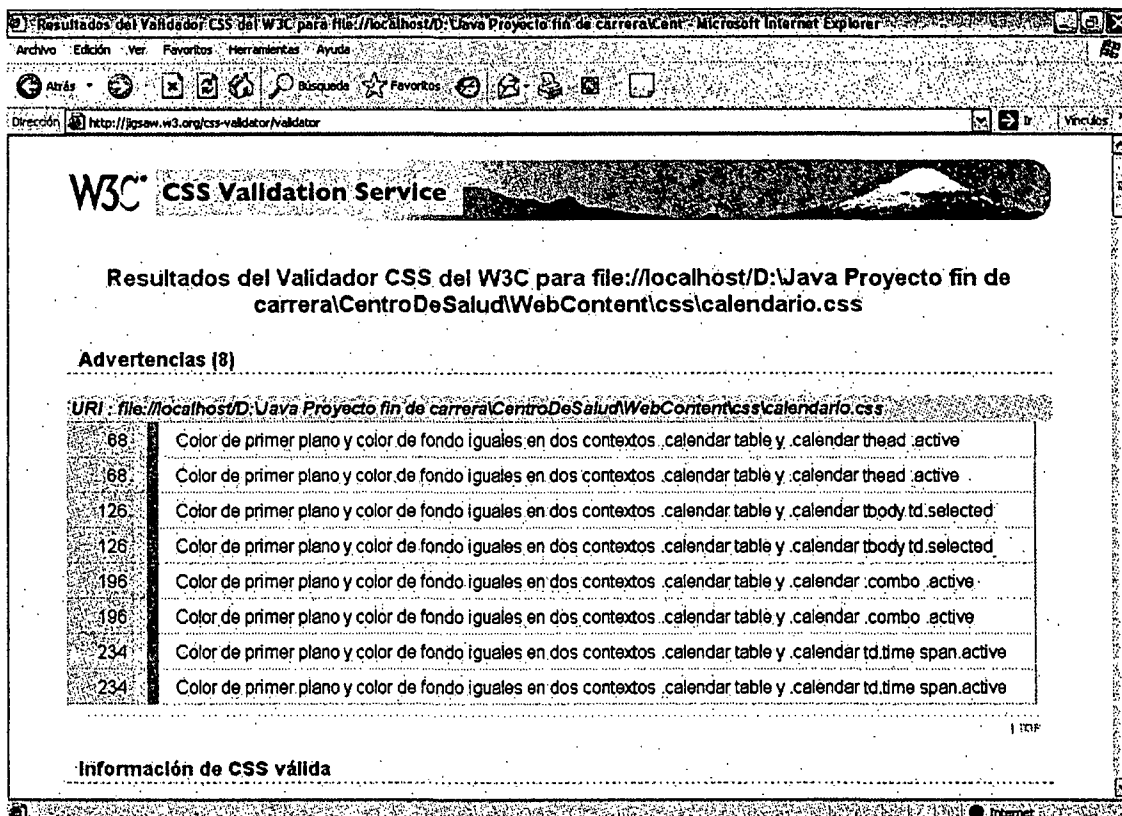
Finalmente es necesario pulsar sobre el botón “Check” para validar que todo sea correcto, apareciendo por consiguiente el resultado.

En este caso, el resultado es correcto, por lo que es una hoja de estilo en cascada correctamente construida.



- Figura 226: Resultado validación de estilos.css -

En el proyecto se usa también otra hoja de estilos para dar formato al calendario, esta hoja tras analizarla se puede observar que contiene errores. Los errores marcados hacen referencia a colores, los cuales se solucionan cambiando estos colores por otros diferentes. En esta versión no se han modificado pues se trata de un componente externo.



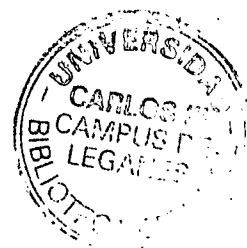
- Figura 227: Resultado validación de calendario.css -

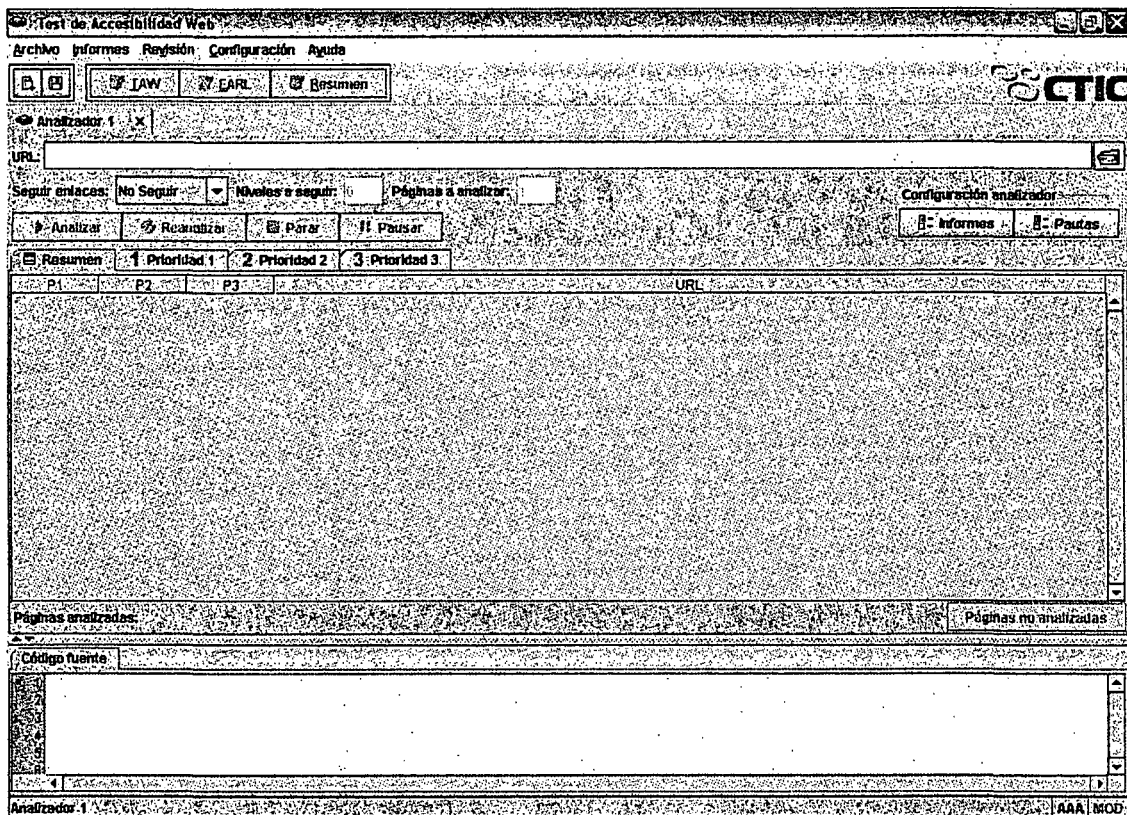
6.3.2.- ANÁLISIS DE LAS PÁGINAS DE ACCESO PÚBLICO

Para realizar el análisis de accesibilidad a las páginas de acceso público, se ha utilizado el programa **Test de Accesibilidad Web 3.08 (TAW)**, desarrollado por el ministerio de Industria, Turismo y Comercio; el Gobierno del Principado de Asturias y la fundación CTIC.

Este programa se puede descargar de forma gratuita a través de la siguiente dirección <http://www.tawdis.net/taw3/cms/es>, aunque también puede usarse de manera on-line.

Tras abrir el programa aparecerá la siguiente pantalla.

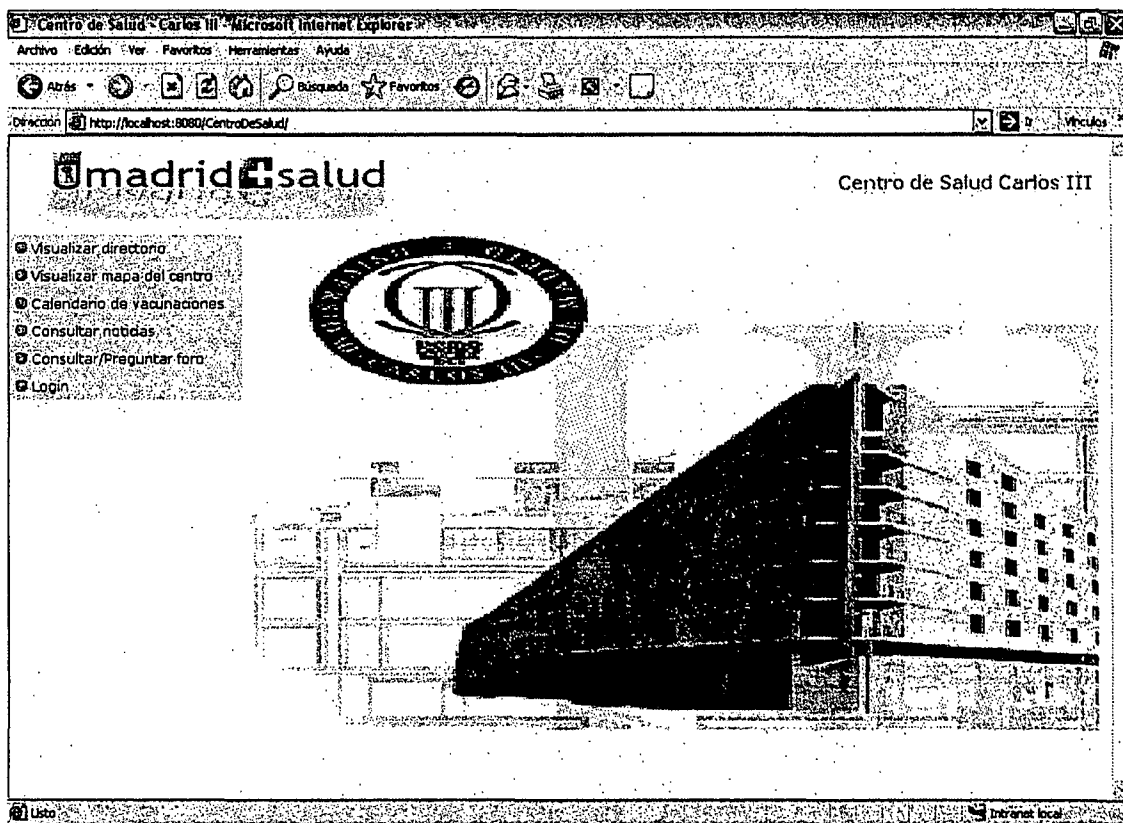




- Figura 228: Test de accesibilidad -

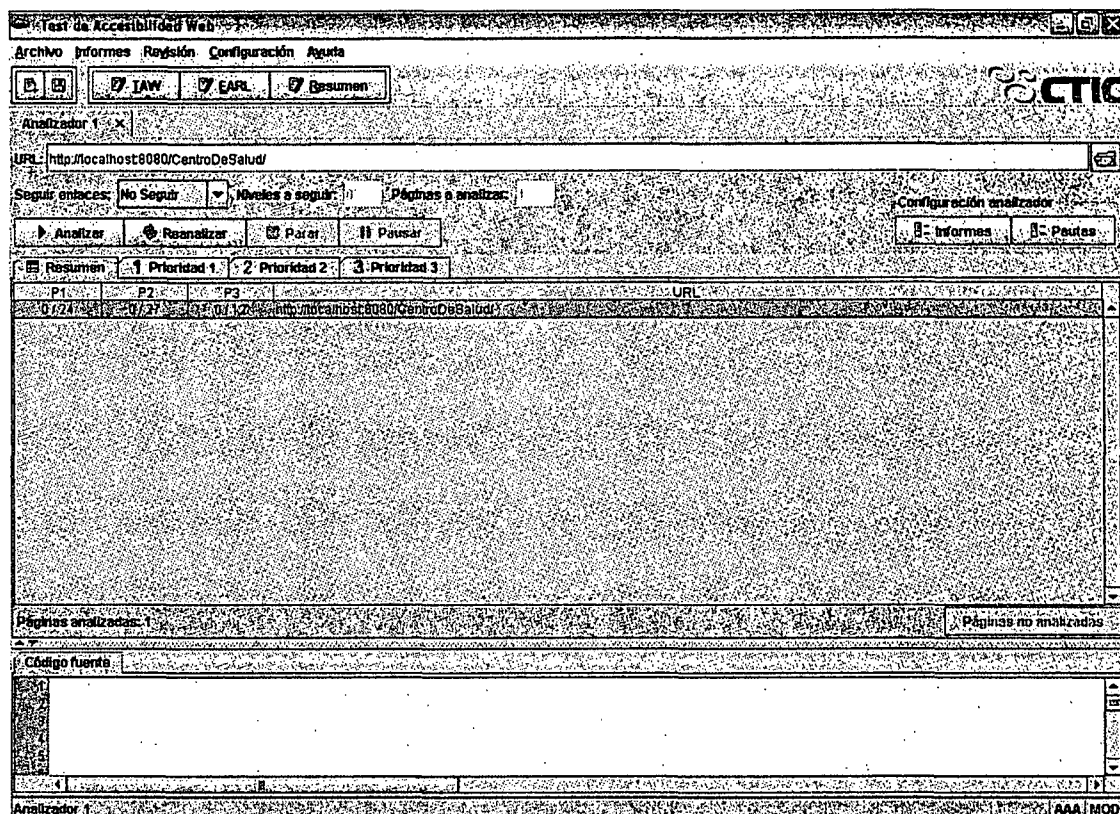
Para analizar cualquier página, solo se necesita introducir la ruta de dicha página ó la ruta del archivo en formato .html, en "URL" y pulsar sobre el botón "Analizar", de forma que debajo aparecerán los resultados, y en cada pestaña se verán desglosados estos, por nivel de accesibilidad.

6.3.2.1.- ANÁLISIS DE LA PÁGINA DE BIENVENIDA



- Figura 229: Bienvenida -

Introducir la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/> y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.

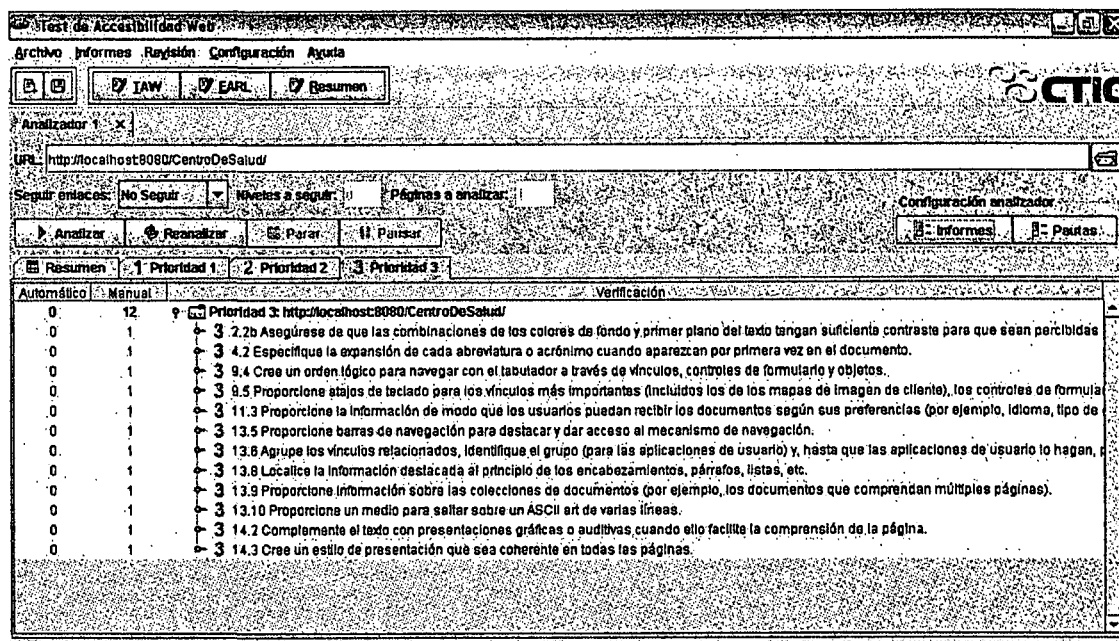


- Figura 230: Resultado accesibilidad Pág. de bienvenida -

Como se puede observar en la imagen, solo existen errores manuales, es decir, aquellos que no puede comprobar el programa, pero que, tras comprobarlo, si se cumplen en la página. En cambio errores automáticos no aparece ninguno, por lo tanto esta página es accesible con prioridad 3, es decir, cumple la triple A.

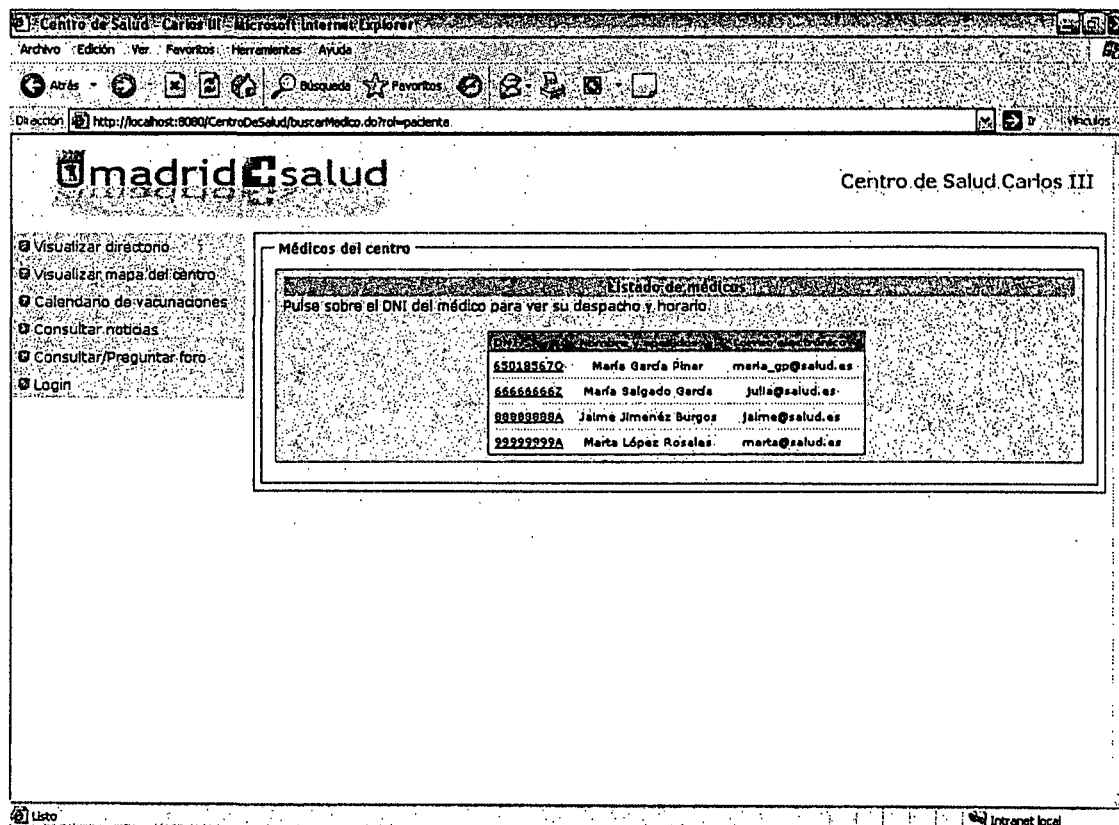
Como ejemplo, se muestra en este caso, la información contenida en las pestañas de prioridad.





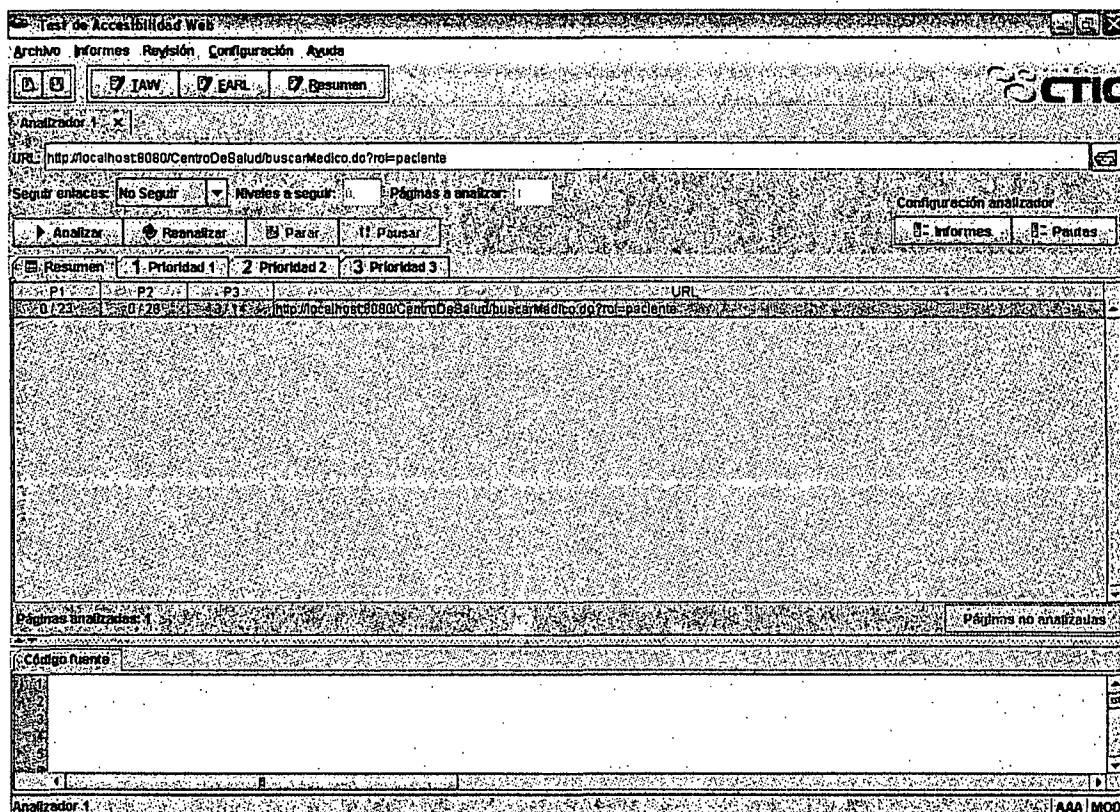
- Figura 233: Resultado accesibilidad Pág. de bienvenida prioridad 3 -

6.3.2.2.- ANÁLISIS DE LA PÁGINAS DE VISUALIZACIÓN DEL DIRECTORIO DE MÉDICOS



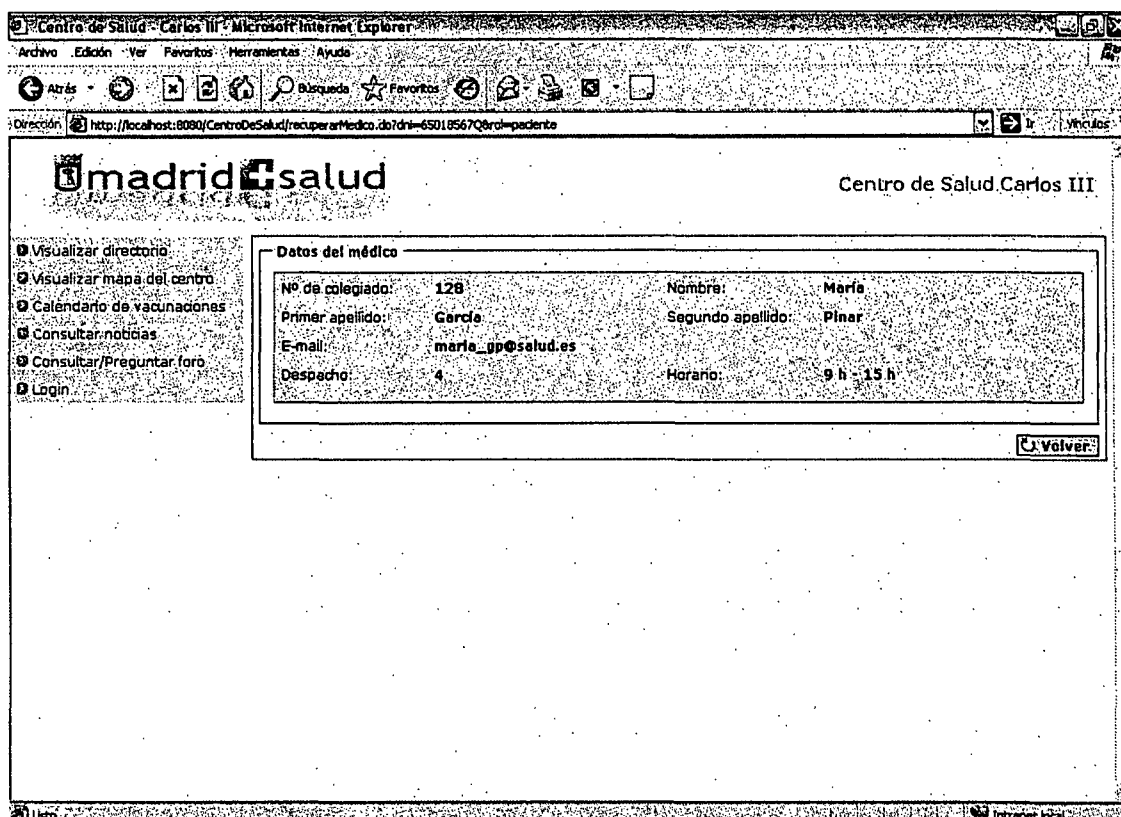
- Figura 234: Visualizar directorio -

Para analizar la primera página, introducir la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/buscarMedico.do?rol=paciente> y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



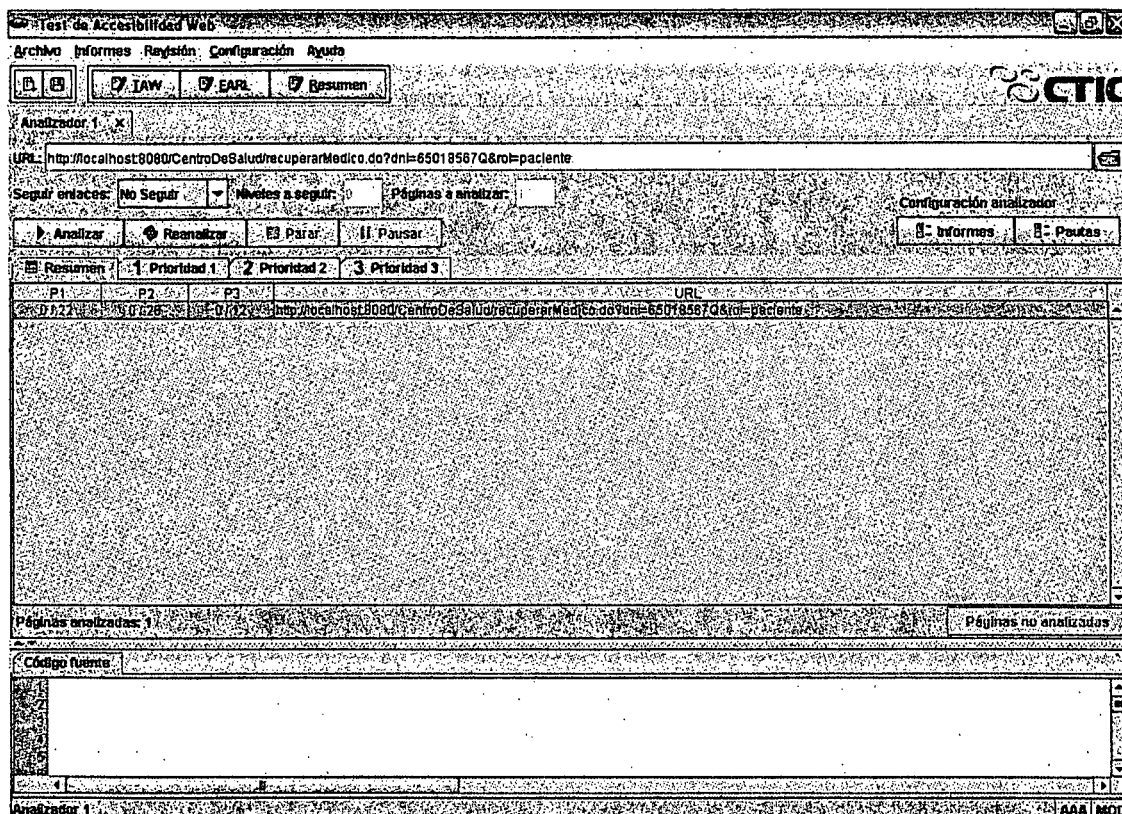
- Figura 235: Resultado accesibilidad Pág. visualizar directorio -

En este caso, la página solo cumple la doble A, ya que tiene dos errores automáticos en la prioridad 3. Sobre estos errores no se hará ningún comentario ya que el cometido de estas páginas es que cumplan la doble A.



- Figura 236: Visualizar datos médicos -

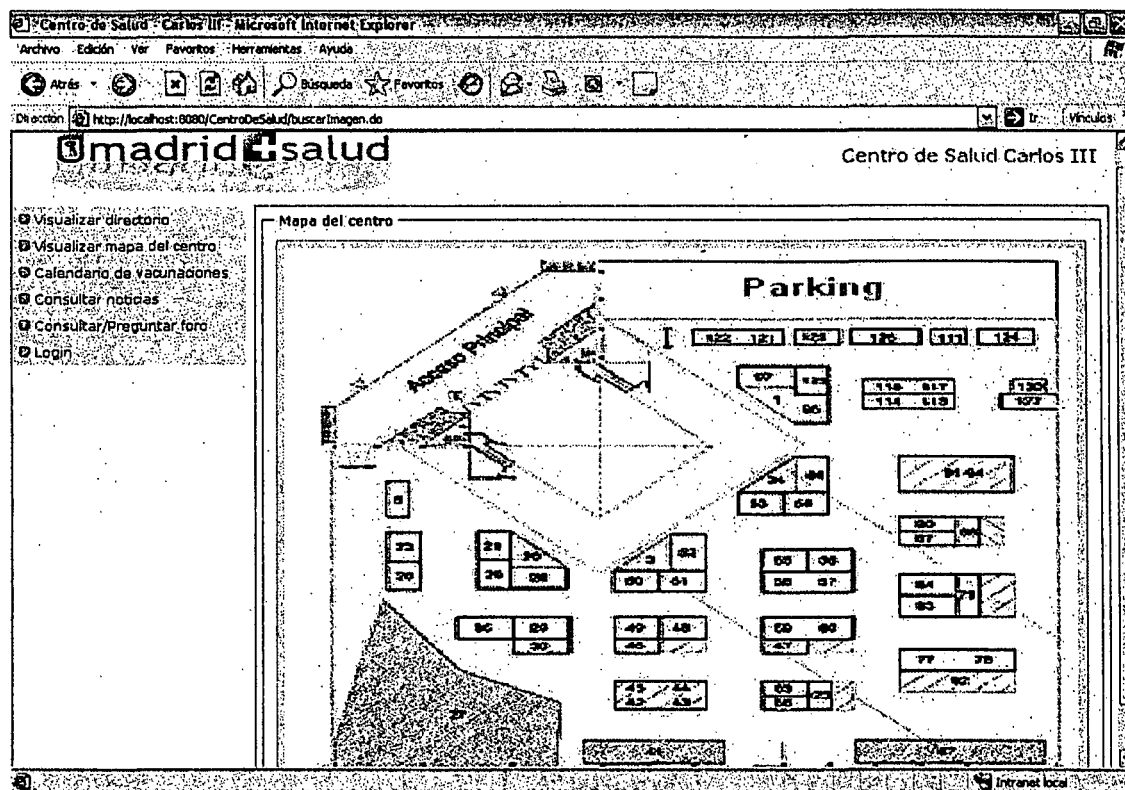
Para analizar la segunda página, introducir la siguiente ruta **http://<host>:<puerto>/CentroDeSalud/recuperarMedico.do?dni=65018567Q&rol=paciente** y pulsar sobre "Analizar". Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 237: Resultado accesibilidad Pág. visualizar datos médicos -

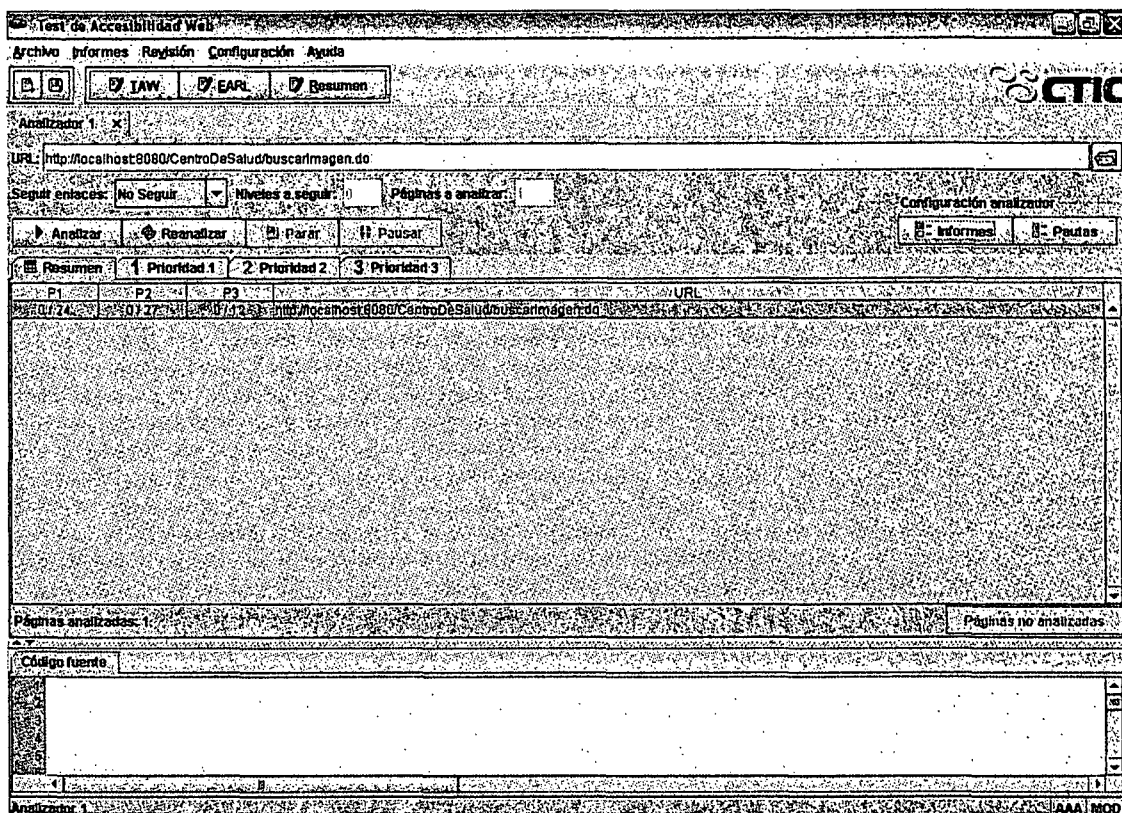
Esta página solo contiene errores manuales, por ello, podemos decir que cumple la tripe A, ya que, tras comprobarlos, si se cumplen en página.

6.3.2.3.- ANÁLISIS DE LA PÁGINA VISUALIZAR MAPA DEL CENTRO



- Figura 238: Visualizar mapa del centro -

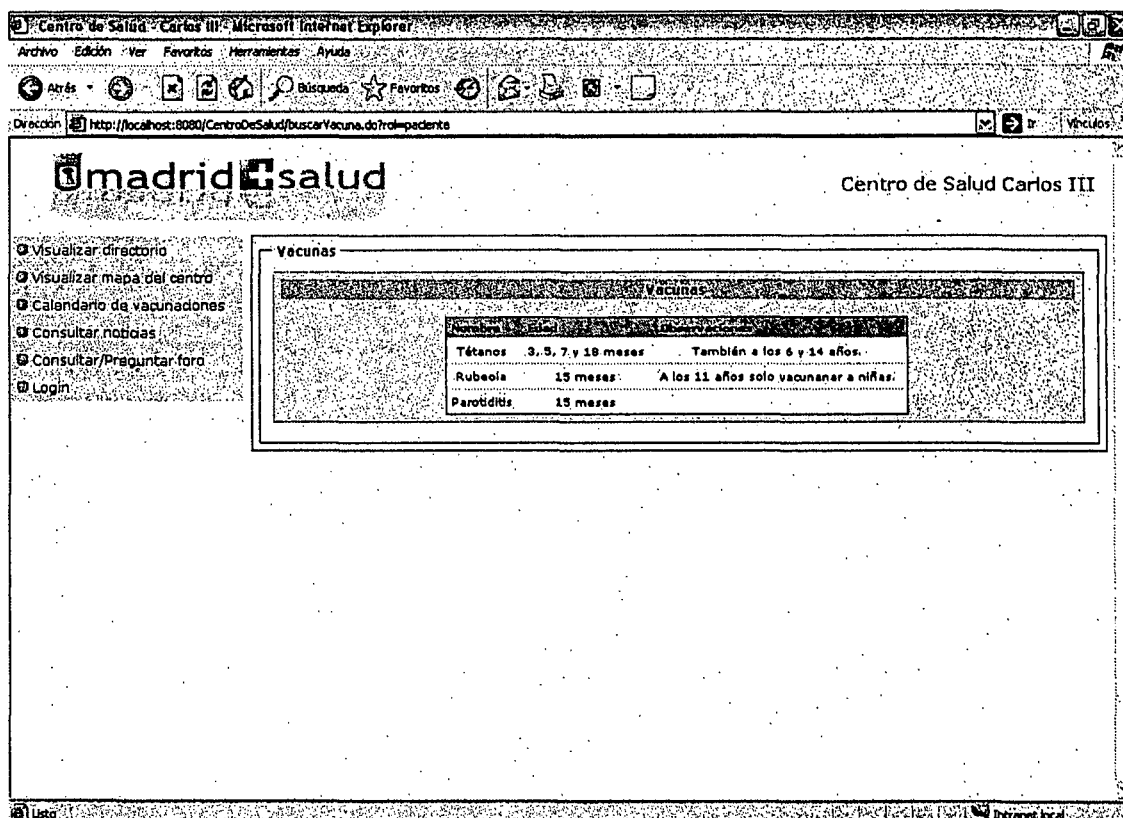
Para analizar la página, introducir la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/buscarImagen.do> y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 239: Resultado accesibilidad Pág. visualizar mapa del centro -

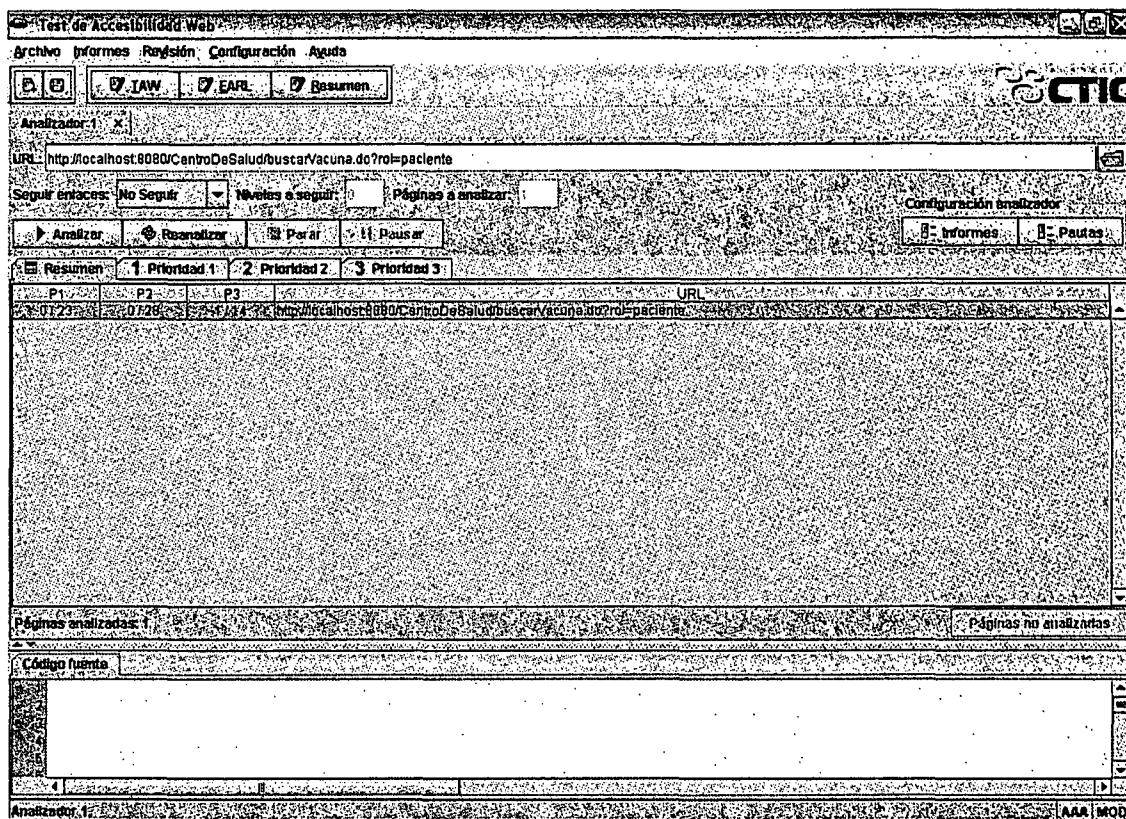
Esta página solo contiene errores manuales, por ello, podemos decir que cumple la tripe A, ya que, tras comprobarlos, si se cumplen en página.

6.3.2.4.- ANÁLISIS DE LA PÁGINA CALENDARIO DE VACUNACIONES



- Figura 240: Calendario de vacunaciones -

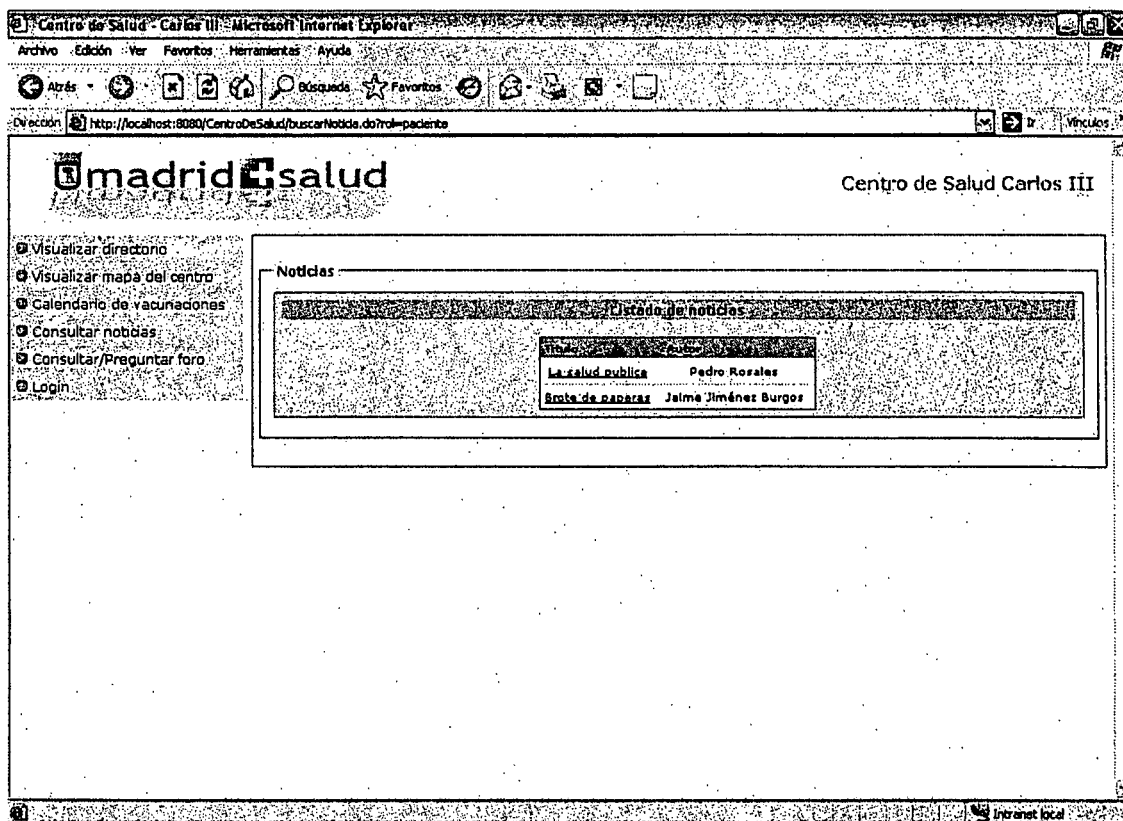
Para analizar la página, introducir la siguiente ruta `http://<host>:<puerto>/CentroDeSalud/buscarVacuna.do?rol=paciente` y pulsar sobre "Analizar". Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 241: Resultado accesibilidad Pág. calendario de vacunaciones -

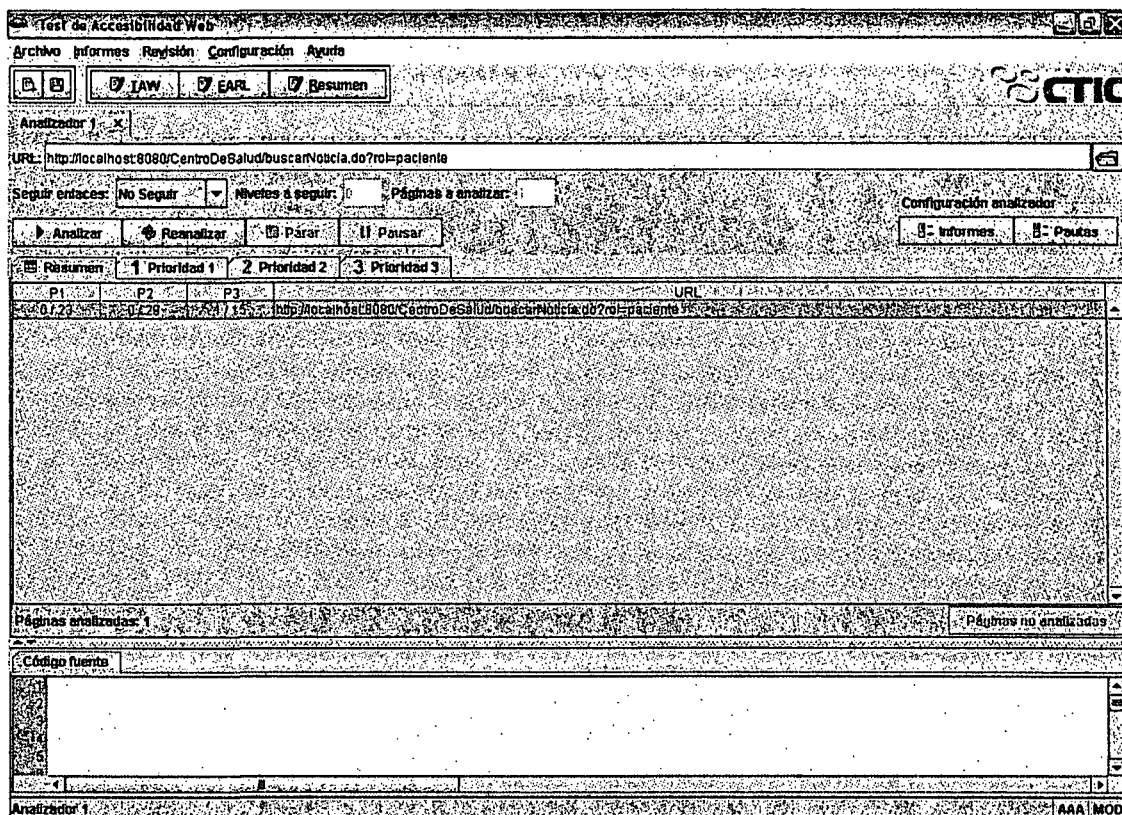
En este caso, la página solo cumple la doble A, ya que tiene un error automático en la prioridad 3.

6.3.2.5.- ANÁLISIS DE LAS PÁGINAS DE CONSULTAR NOTICIAS



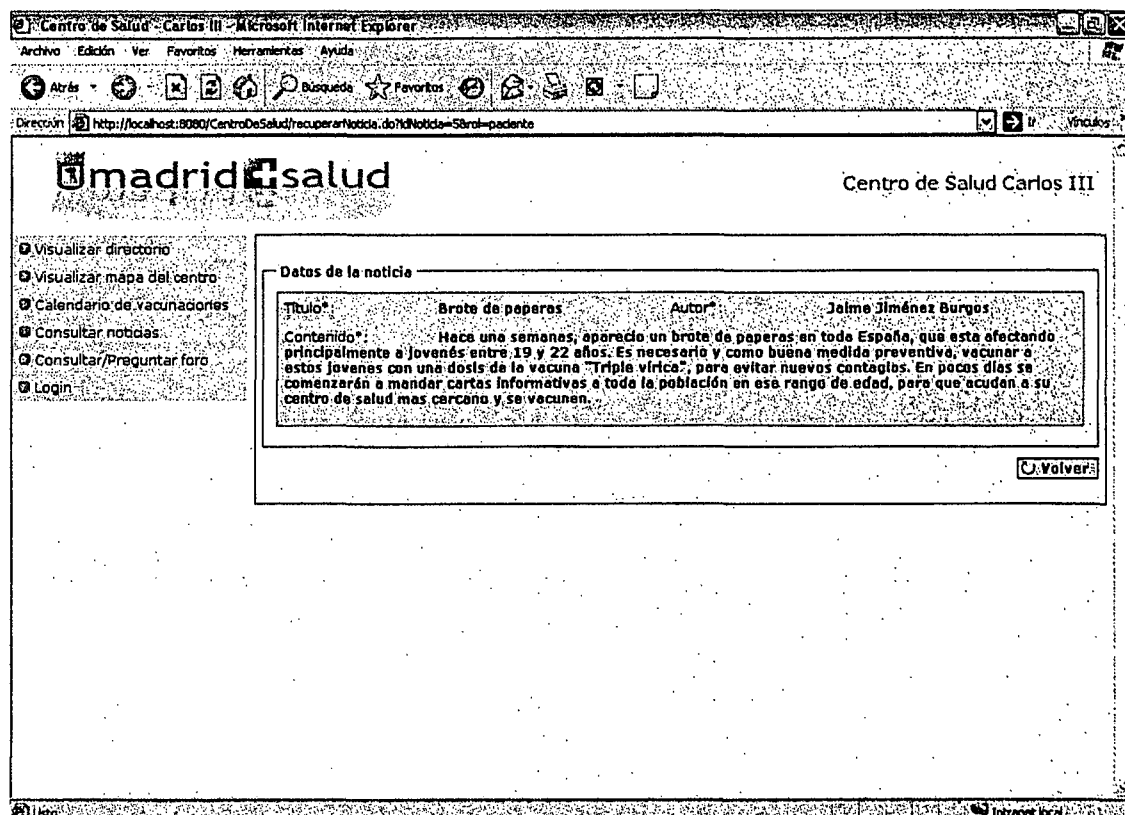
- Figura 242: Consultar noticias -

Para analizar la primera página, introducir la siguiente ruta **http://<host>:<puerto>/CentroDeSalud/buscarNoticia.do?rol=paciente** y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



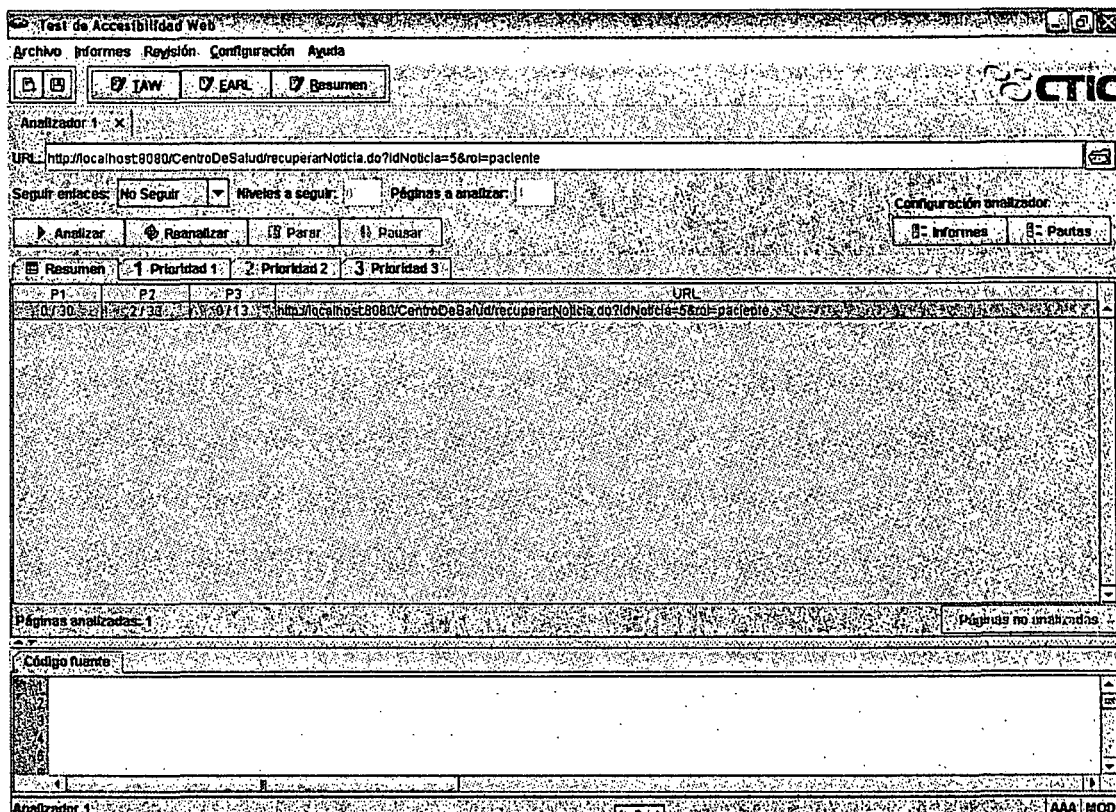
- Figura 243: Resultado accesibilidad Pág. consultar noticias-

En este caso, la página solo cumple la doble A, ya que contiene un error automático en la prioridad 3.



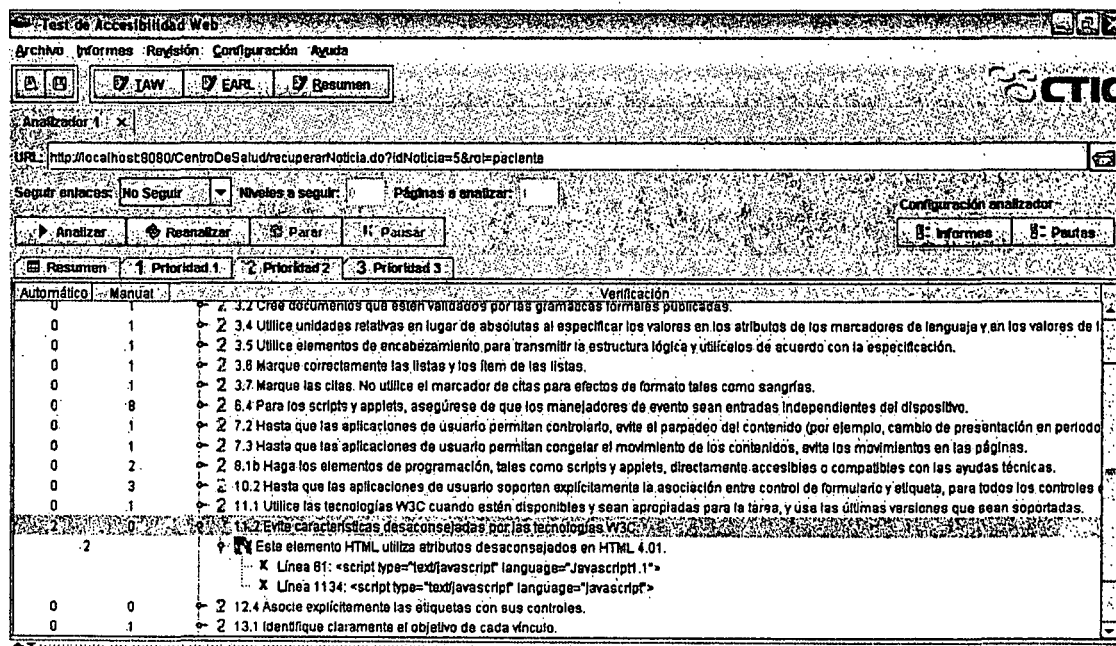
- Figura 244: Visualizar noticia -

Para analizar la segunda página, introducir la siguiente ruta **http://<host>:<puerto>/CentroDeSalud/recuperarNoticia.do?idNoticia=5&rol=paciente** y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 245: Resultado accesibilidad Pág. visualizar noticia -

En esta página aparecen dos errores automáticos de prioridad 2, con lo cual, solo cumple A.

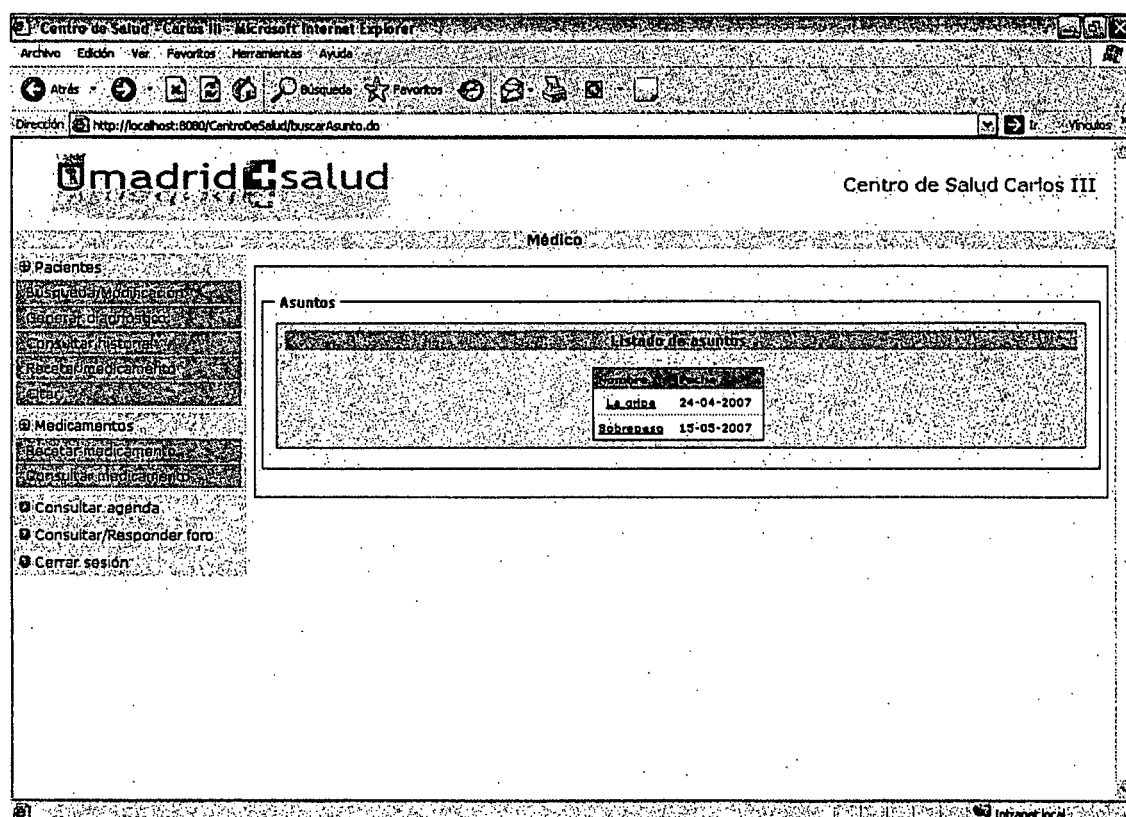


- Figura 246: Resultado accesibilidad Pág. visualizar noticia prioridad 2 -

Tras comprobar los errores, se observa que se corresponde con la pauta 11, e indican que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador del usuario no se podría completar la funcionalidad.

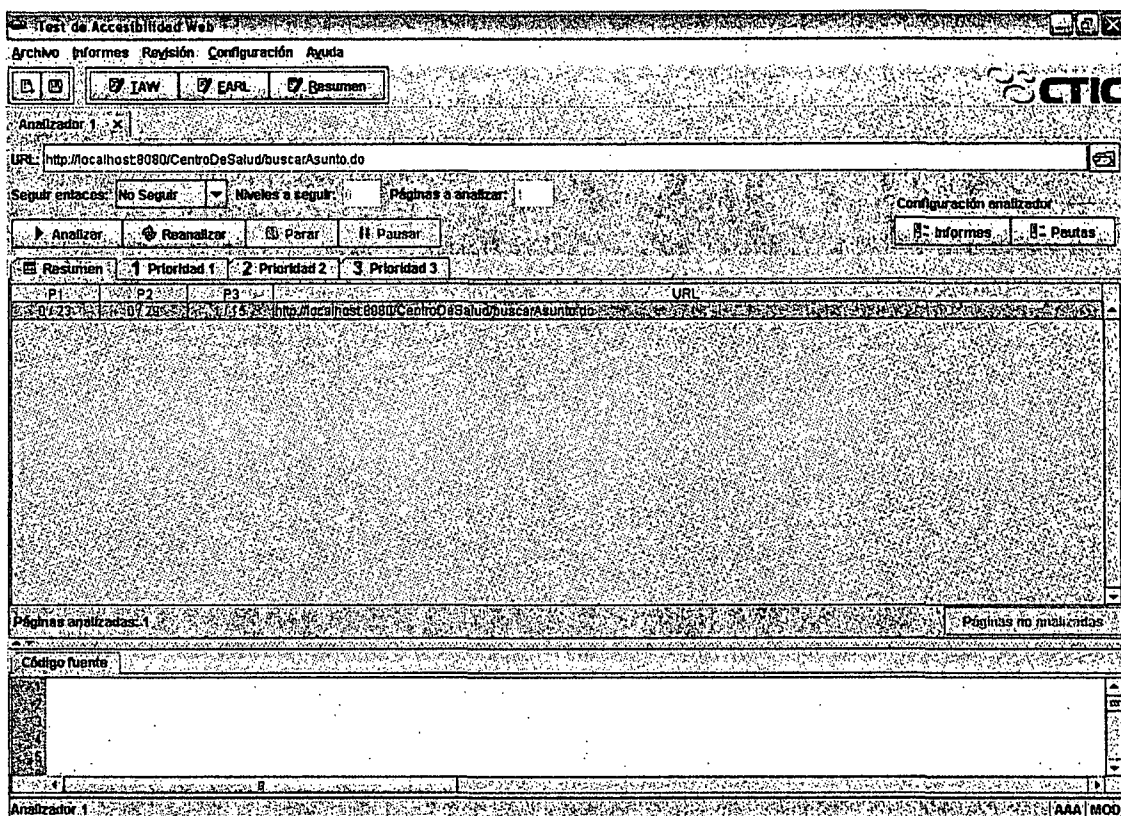
En este caso, esta página es la misma que se utiliza cuando un administrador da de alta noticias, por lo tanto el primer elemento JavaScript hace referencias a las validaciones de campos, mientras que el segundo se utiliza para confirmar la eliminación de alguna noticia. Por ello, esto no afecta al usuario, de modo que se cumple la triple A.

6.3.2.6.- ANÁLISIS DE LAS PÁGINAS DEL FORO



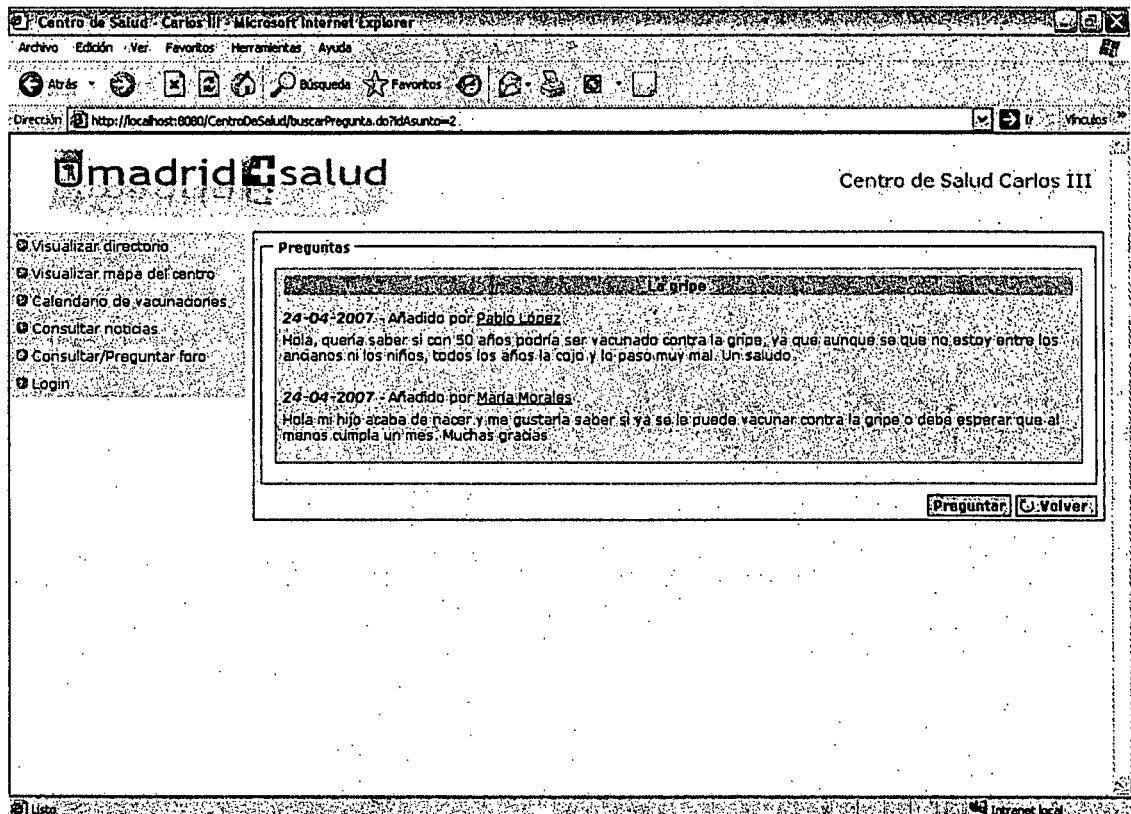
- Figura 247: Listado de asuntos -

Para analizar la primera página, introducir la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/buscarAsunto.do> y pulsar sobre "Analizar". Tras realizar el análisis aparecerán los siguientes resultados.



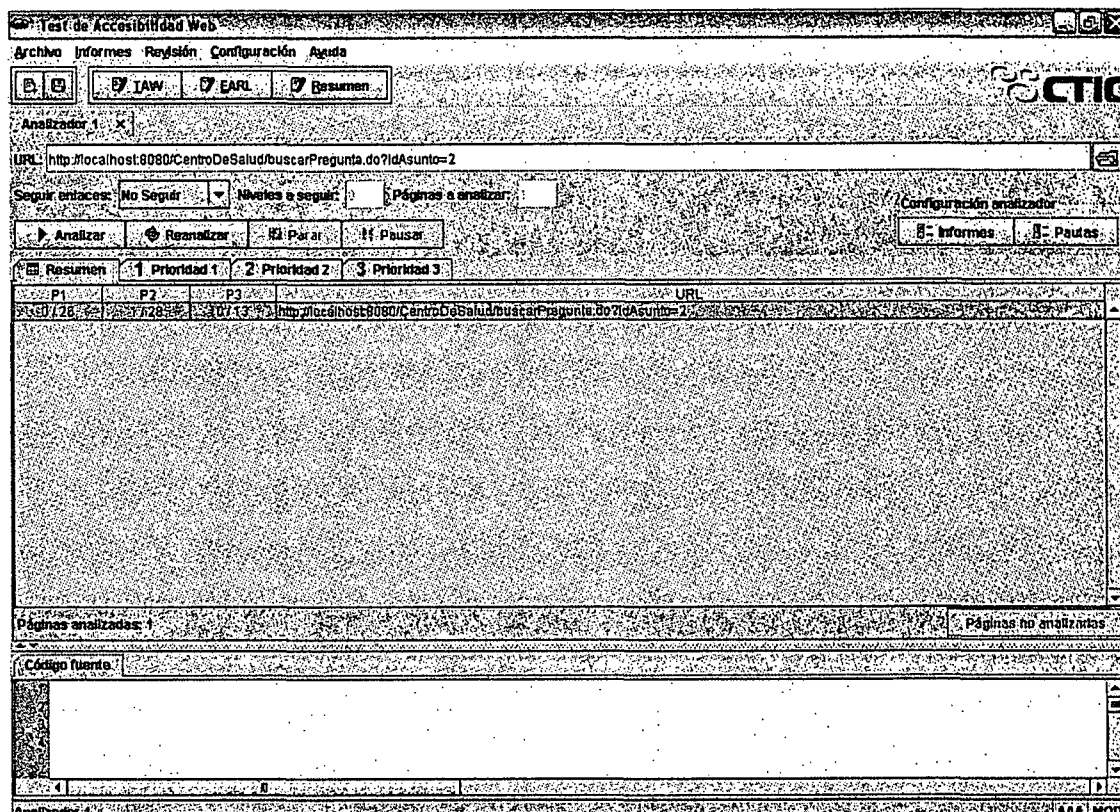
- Figura 248: Resultado accesibilidad Pág. listado de asuntos -

En este caso, la página solo cumple la doble A, ya que contiene un error automático en la prioridad 3.



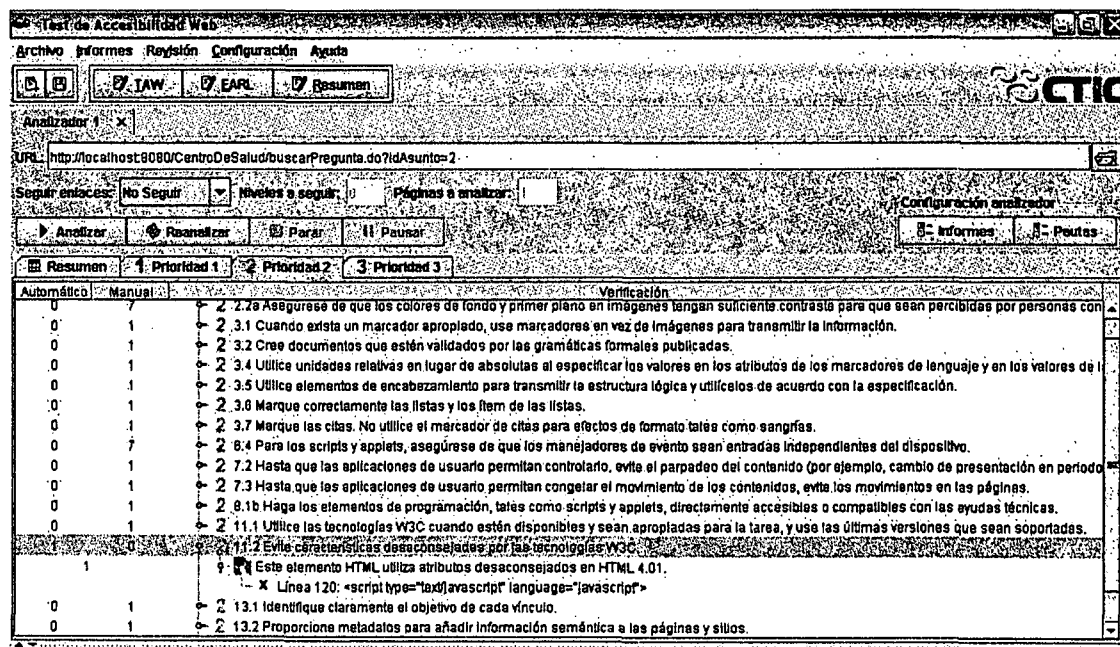
- Figura 249: Visualizar preguntas -

Para analizar la segunda página, introducir la siguiente ruta **http://<host>:<puerto>/CentroDeSalud/buscarPregunta.do?idAsunto=2** y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 250: Resultado accesibilidad Pág. visualizar preguntas -

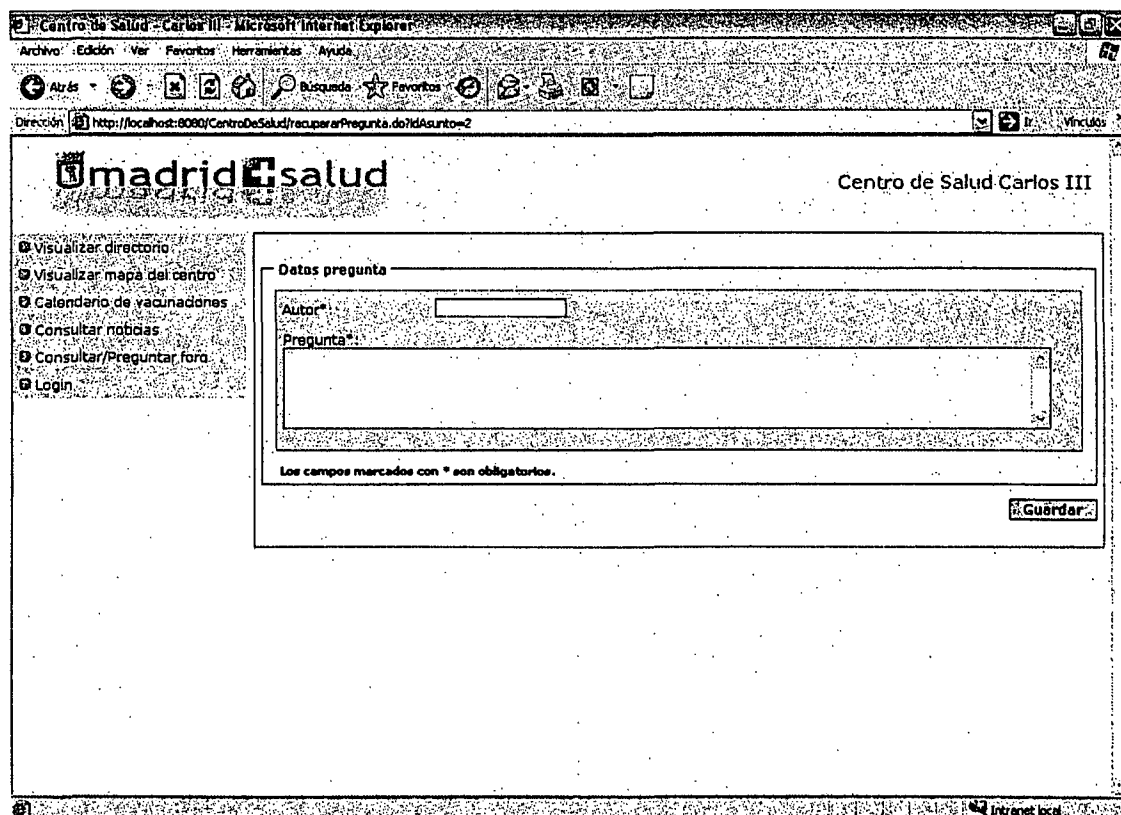
En esta página aparece un error automático de prioridad 2, con lo cual, solo cumple A.



- Figura 251: Resultado accesibilidad Pág. visualizar preguntas prioridad 2 -

Tras comprobar el error, se observa que se corresponde con la pauta 11, e indica que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador del usuario no se podría completar la funcionalidad.

En este caso esta página es usada tanto por el médico y por el administrador para consultar el foro, por lo tanto el elemento JavaScript es usado para confirmar la eliminación de un asunto por parte de un administrador. De esta forma no es aplicable al usuario, por lo tanto, la página cumple la triple A.



Centro de Salud Carlos III - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atás Busqueda Favoritos

Dirección <http://localhost:8080/CentroDeSalud/recuperarPregunta.do?idAsunto=2> Vínculos

madrid salud Centro de Salud Carlos III

- Visualizar directorio
- Visualizar mapa del centro
- Calendario de vacunaciones
- Consultar noticias
- Consultar/Preguntar foro
- Login

Datos pregunta

Autor*

Pregunta*

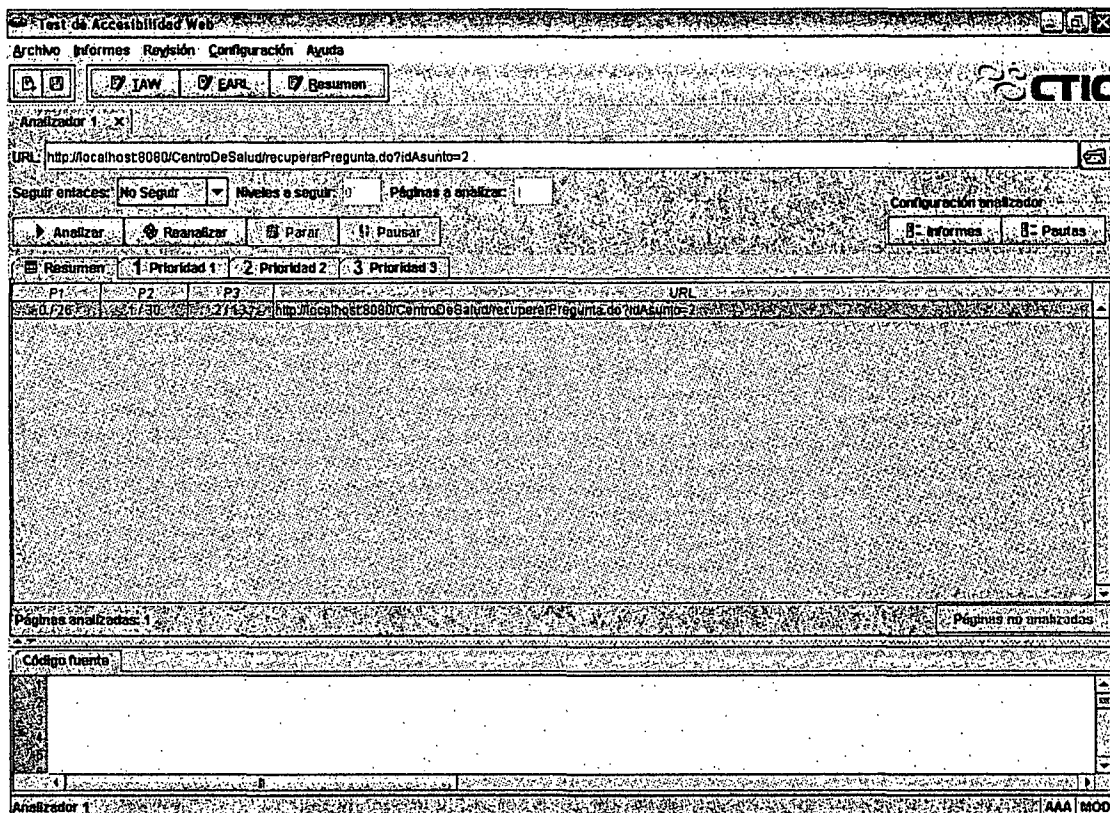
Los campos marcados con * son obligatorios.

Guardar

Intranet local

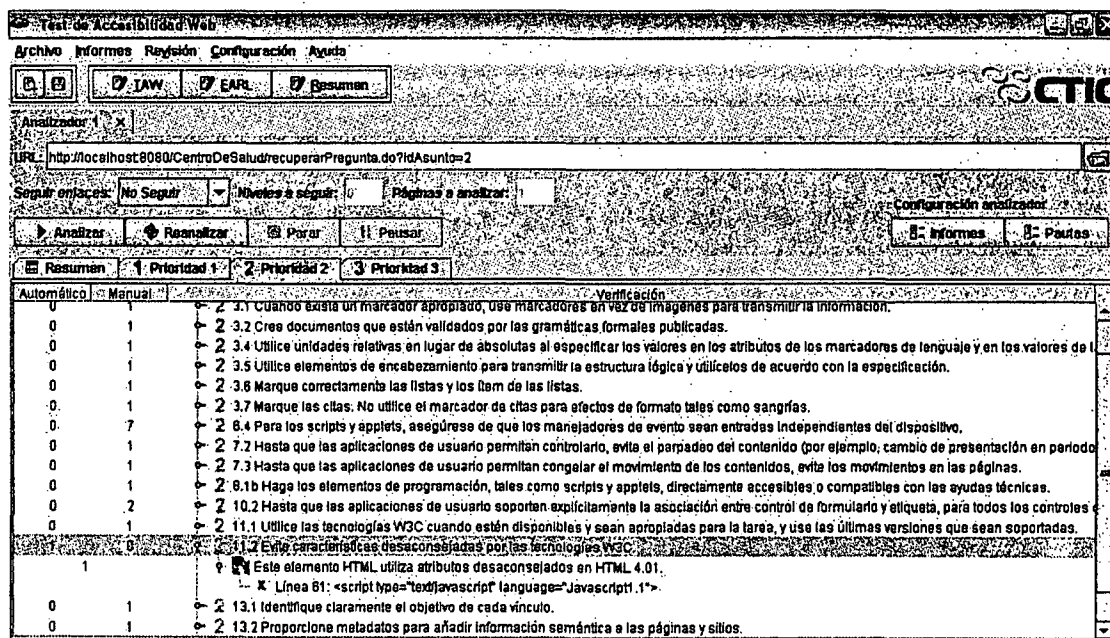
- Figura 252: Preguntar -

Para analizar la tercera página, introducir la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/recuperarPregunta.do?idAsunto=2> y pulsar sobre "Analizar". Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 253: Resultado accesibilidad Pág. preguntar -

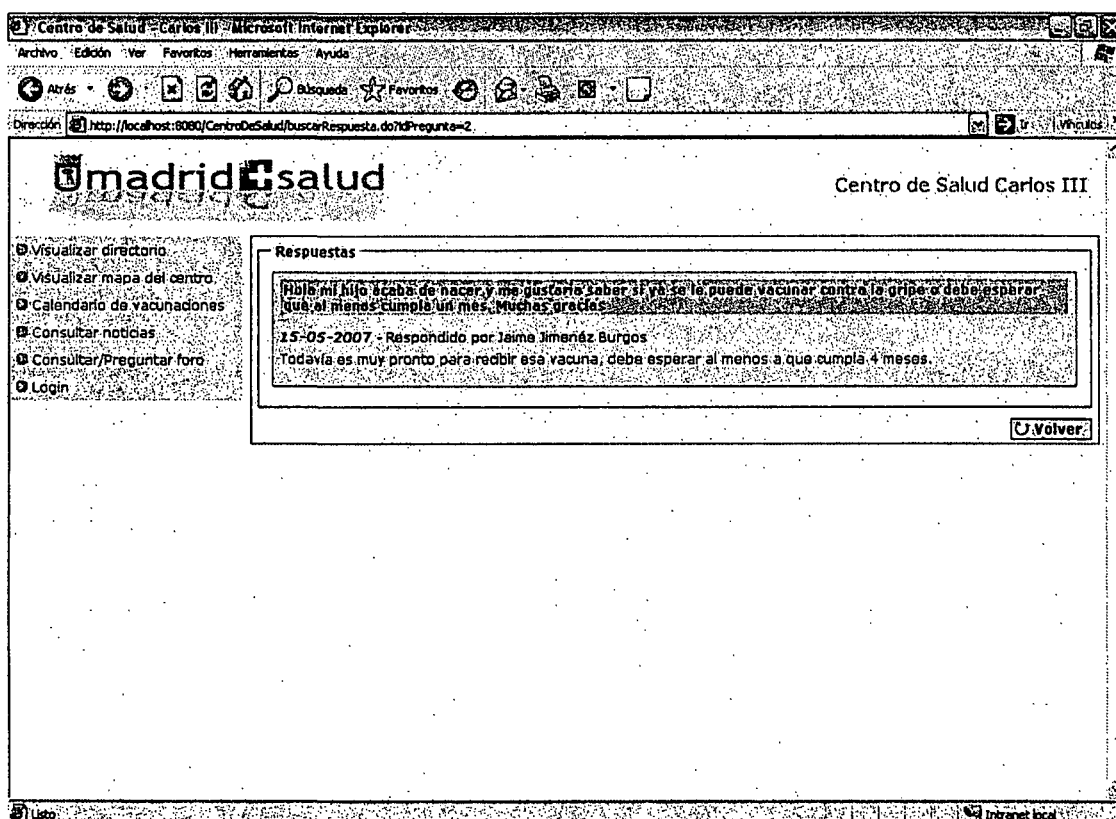
En esta página aparece un error automático de prioridad 2 y dos de prioridad 3, con lo cual, solo cumple A.



- Figura 254: Resultado accesibilidad Pág. preguntar prioridad 2 -

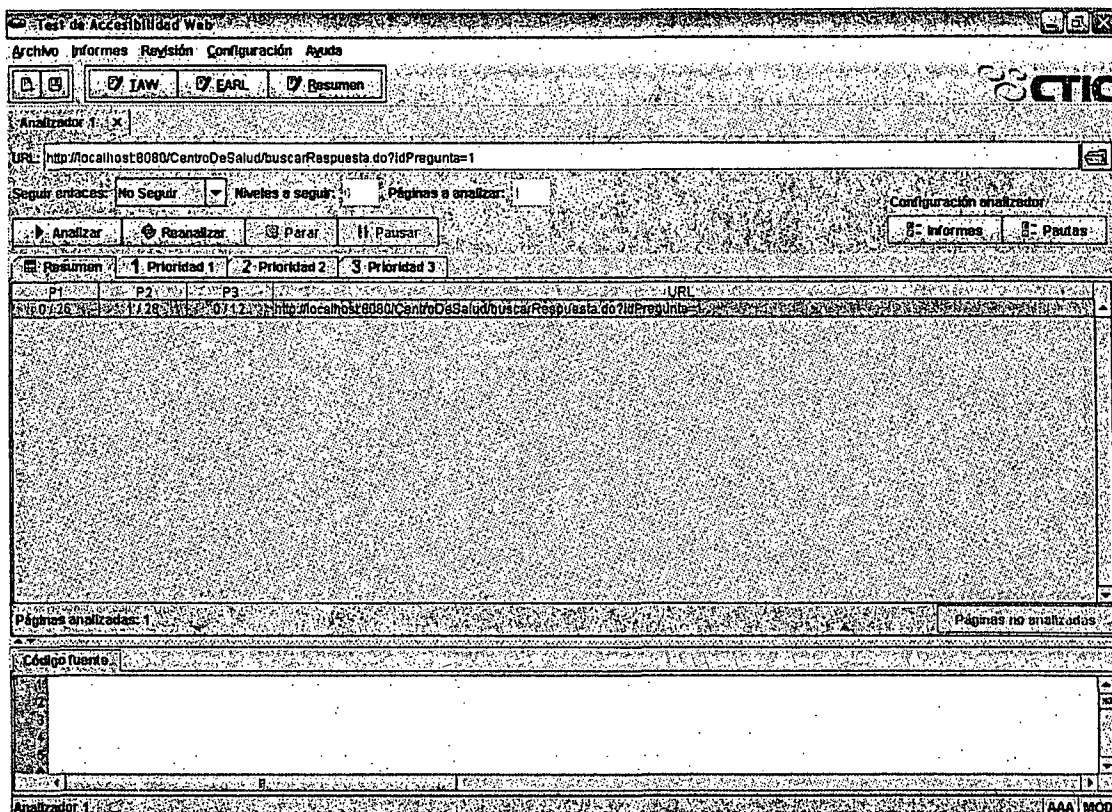
Tras comprobar el error de prioridad 2, se observa que se corresponde con la pauta 11, e indica que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador del usuario no se podría completar la funcionalidad.

En este caso, el elemento JavaScript se usa para validar que el usuario rellene todos los campos que son obligatorios. Estas validaciones están hechas con el marco Validator de Struts, por lo tanto, en caso de que el usuario desactive JavaScript en su navegador, las validaciones se harán en el servidor. Esto indica que no es aplicable ya que sin JavaScript todo funcionará correctamente, por lo tanto esta página cumple la doble A.



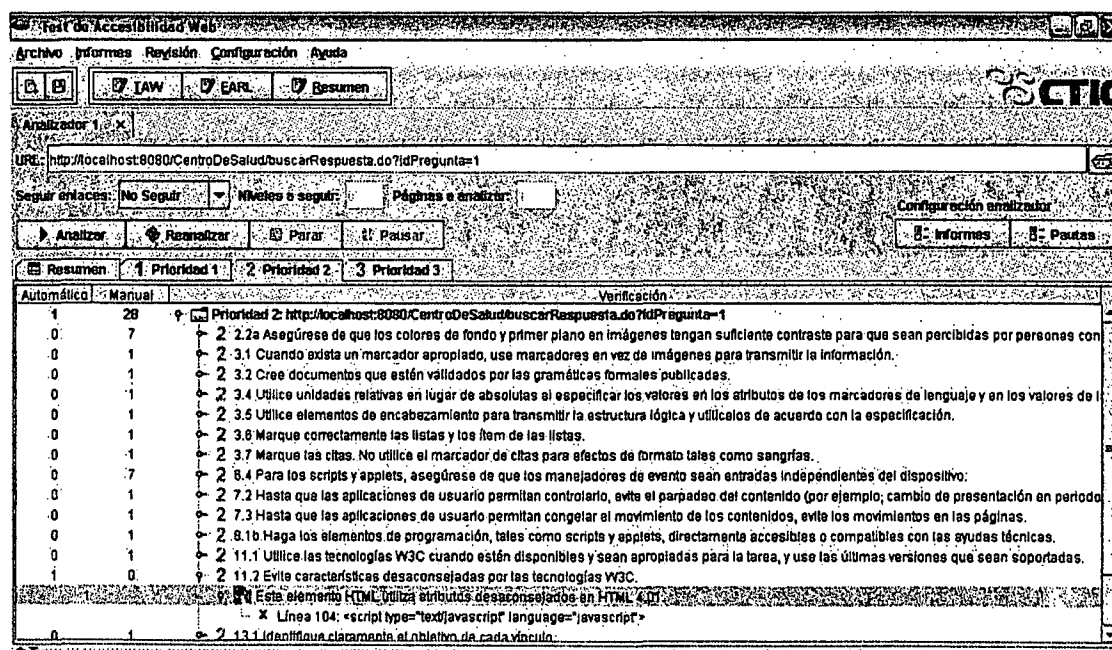
- Figura 255: Visualizar respuesta -

Para analizar la cuarta página, introducir la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/buscarRespuesta.do?idPregunta=1> y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 256: Resultado accesibilidad Pág. visualizar respuestas -

En esta página aparece un error automático de prioridad 2, con lo cual, solo cumple A.

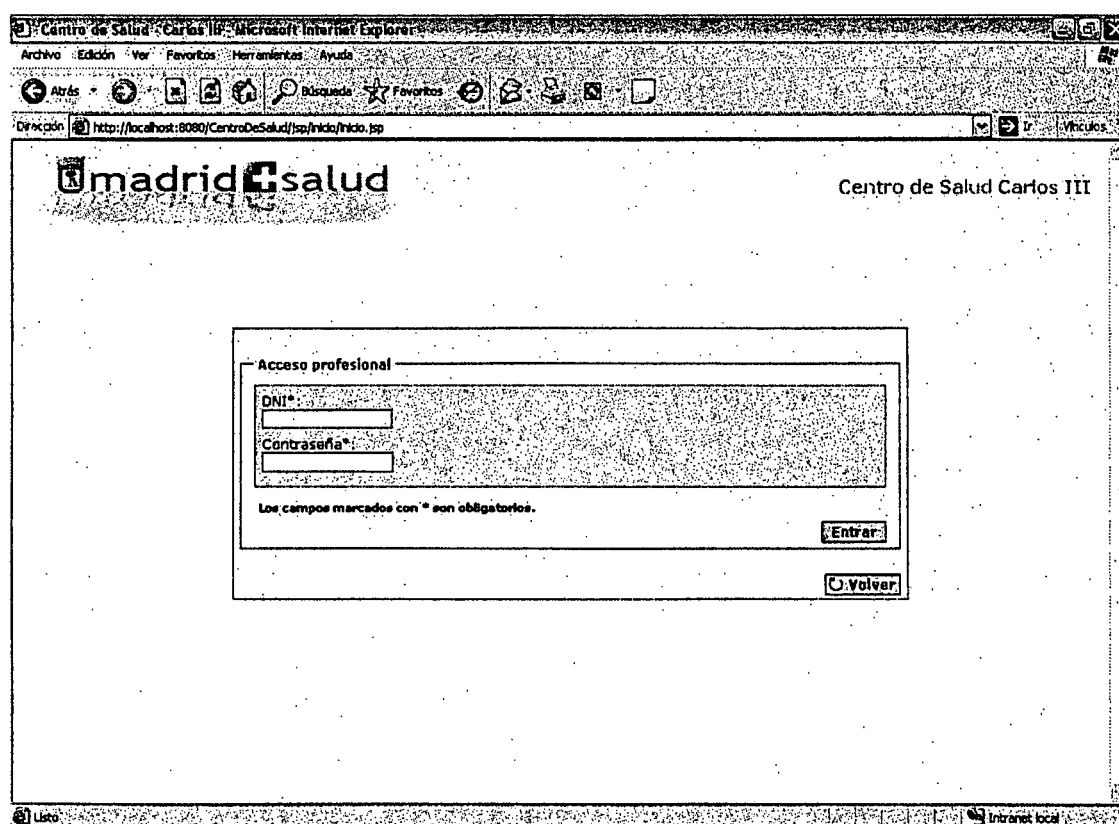


- Figura 257: Resultado accesibilidad Pág. visualizar respuestas prioridad 2 -

Tras comprobar el error se observa que se corresponde con la pauta 11, e indica que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador del usuario no se podría completar la funcionalidad.

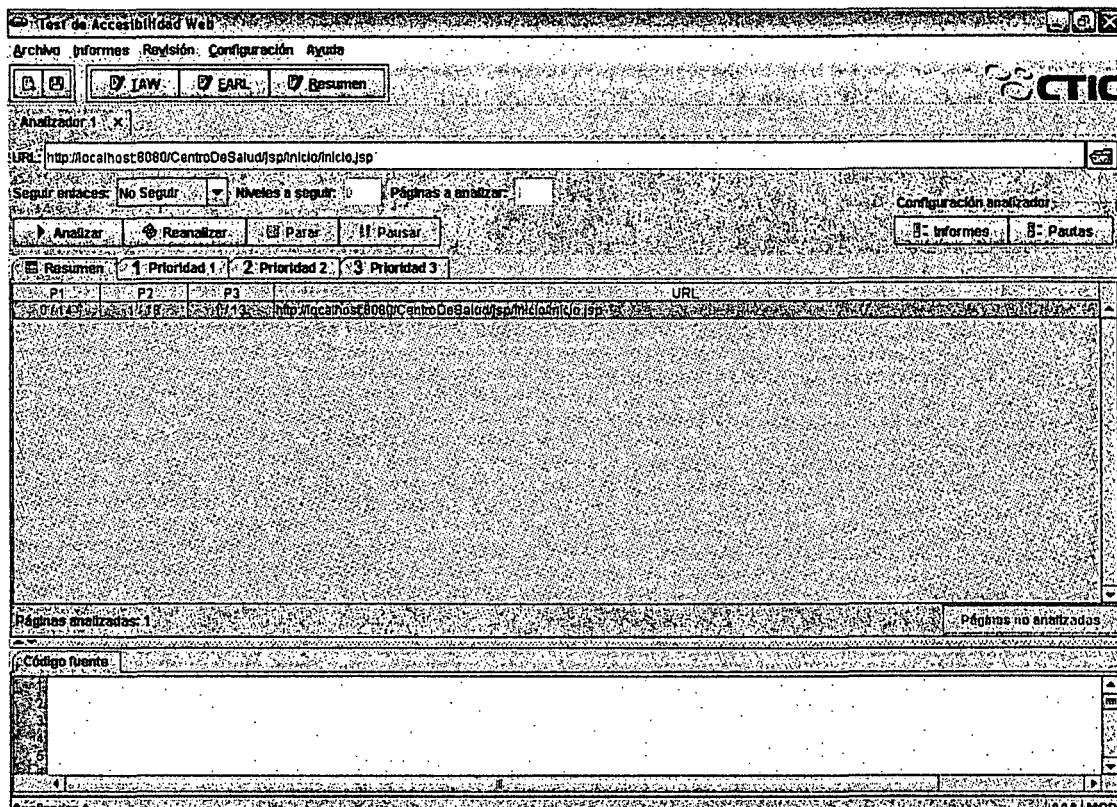
En este caso esta página es usada tanto por el médico y por el administrador para consultar el foro, por lo tanto el elemento JavaScript es usado para confirmar la eliminación de una pregunta por parte de un administrador. De esta forma no es aplicable al usuario, por lo tanto, la página cumple la triple A.

6.3.2.7.- ANÁLISIS DE LA PÁGINA DE LOGIN



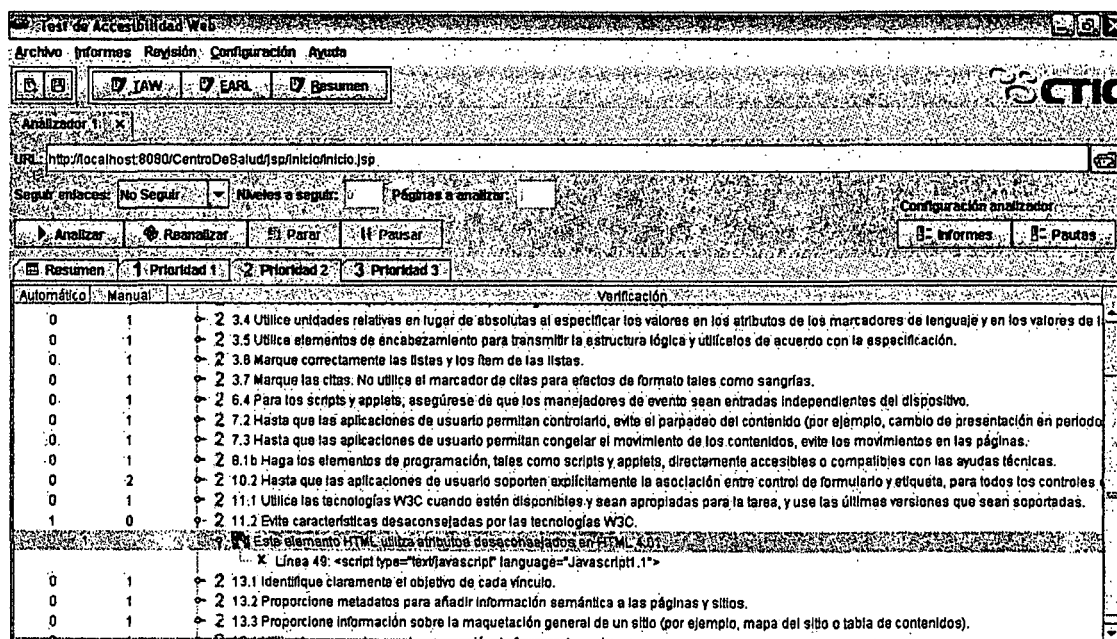
- Figura 258: Login -

Para analizar la página, introducir la siguiente ruta `http://<host>:<puerto>/CentroDeSalud/jsp/Inicio/Inicio.jsp` y pulsar sobre "Analizar". Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 259: Resultado accesibilidad Pág. de login -

En este caso, la página solo cumple la doble A, ya que contiene un error automático en la prioridad 2 y otro en la 3.



- Figura 260: Resultado accesibilidad Pág. de login prioridad 2 -



Tras comprobar el error de prioridad 2, se observa que se corresponde con la pauta 11, e indica que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador del usuario no se podría completar la funcionalidad.

En este caso, el elemento JavaScript se usa para validar que el usuario rellene todos los campos que son obligatorios. Estas validaciones están hechas con el marco Validator de Struts, por lo tanto, en caso de que el usuario desactive JavaScript en su navegador, las validaciones se harán en el servidor. Esto indica que no es aplicable ya que sin JavaScript todo funcionará correctamente, por lo tanto esta página cumple la doble A.

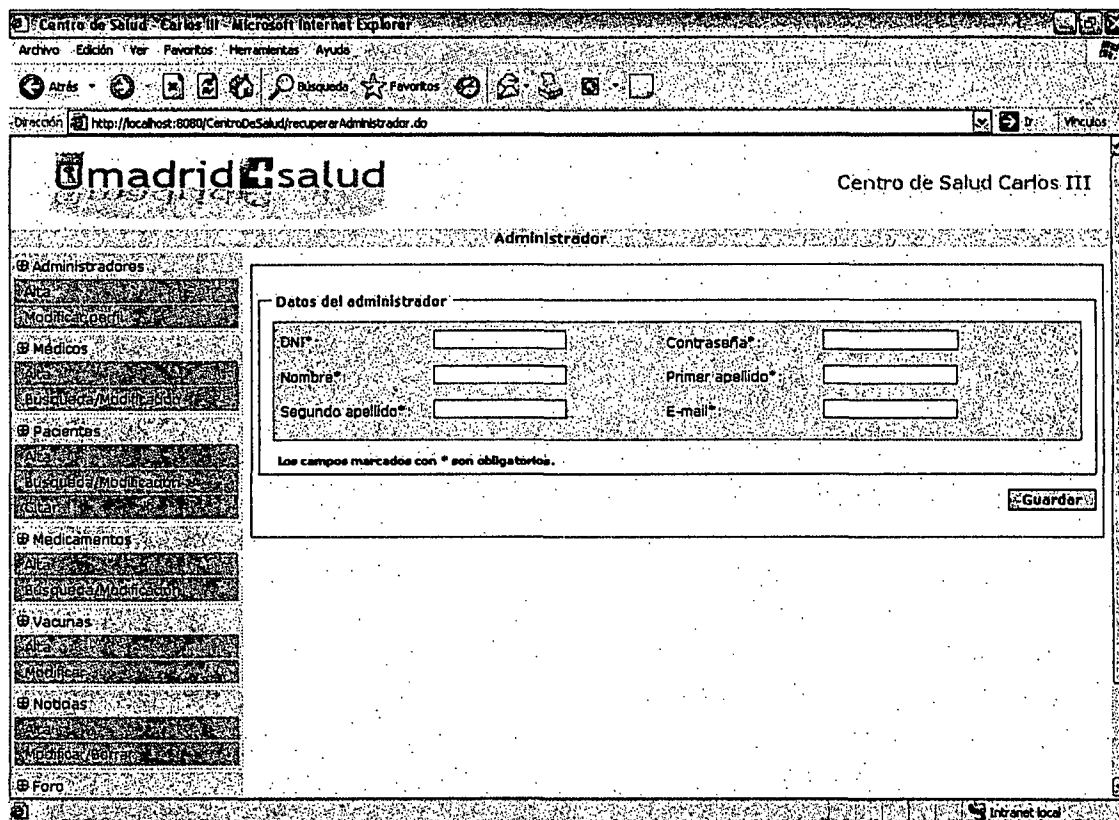
6.3.3.- ANÁLISIS DE LAS PÁGINAS DE ACCESO PRIVADO

En este apartado se analizarán algunas de las páginas de acceso privado de la aplicación, es decir, aquellas a las que solo tienen acceso los médicos y administradores del centro.

Al ser demasiadas páginas las cuales mantienen una estructura común, se ha decidido analizar una página por cada una de las principales funcionalidades (guardar, recuperar, buscar y borrar).

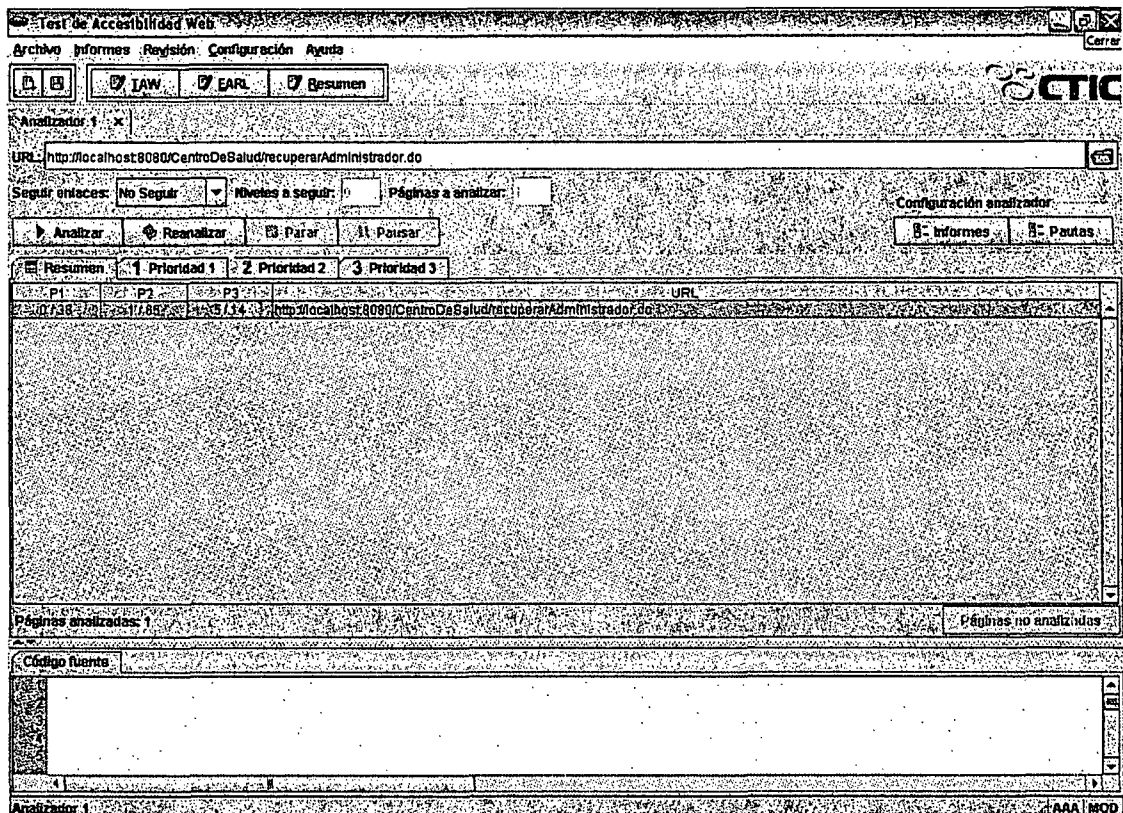
Para estas páginas se ha pretendido que al menos cumplan la A, ya que en algunas de ellas existen elementos JavaScript necesarios para confirmar, por ejemplos, la eliminación de algún elemento, con lo cual, es necesario que mantengan JavaScript activo, de forma que muchas de ellas no podrán cumplir la doble A.

6.3.3.1.- ANÁLISIS DE UNA PÁGINA DE ALTA



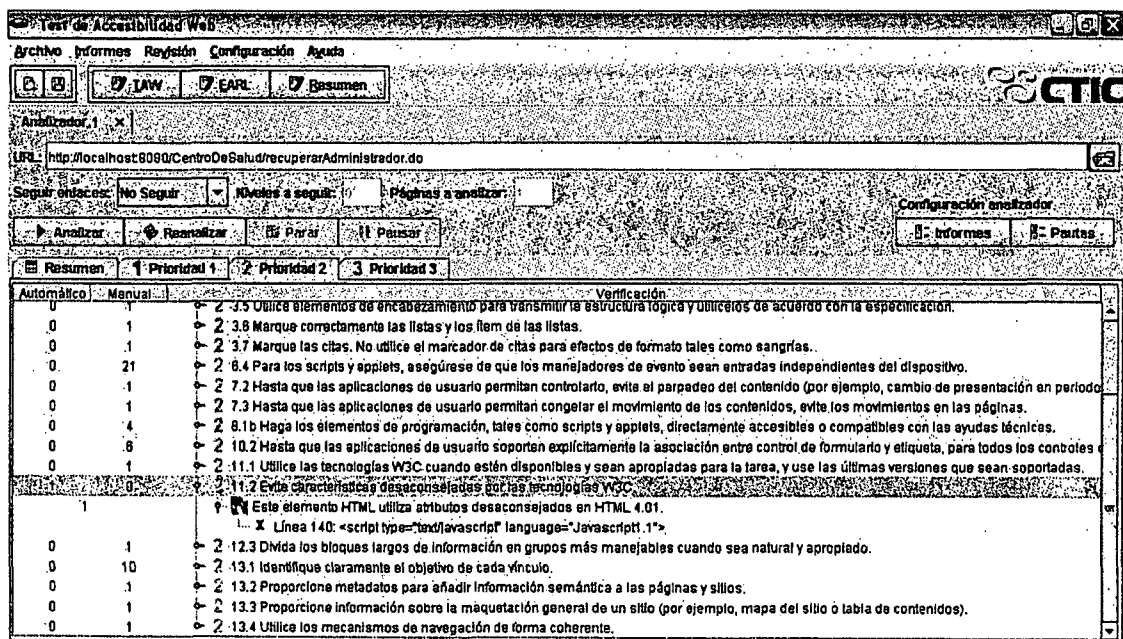
- Figura 261: Alta administrador-

Introducir por ejemplo la siguiente ruta **http://<host>:<puerto>/CentroDeSalud/recuperarAdministrador.do** y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 262: Resultado accesibilidad Pág. alta administrador -

En esta página aparece un error automático de prioridad 2 y cinco de prioridad 3, con lo cual, solo cumple A.

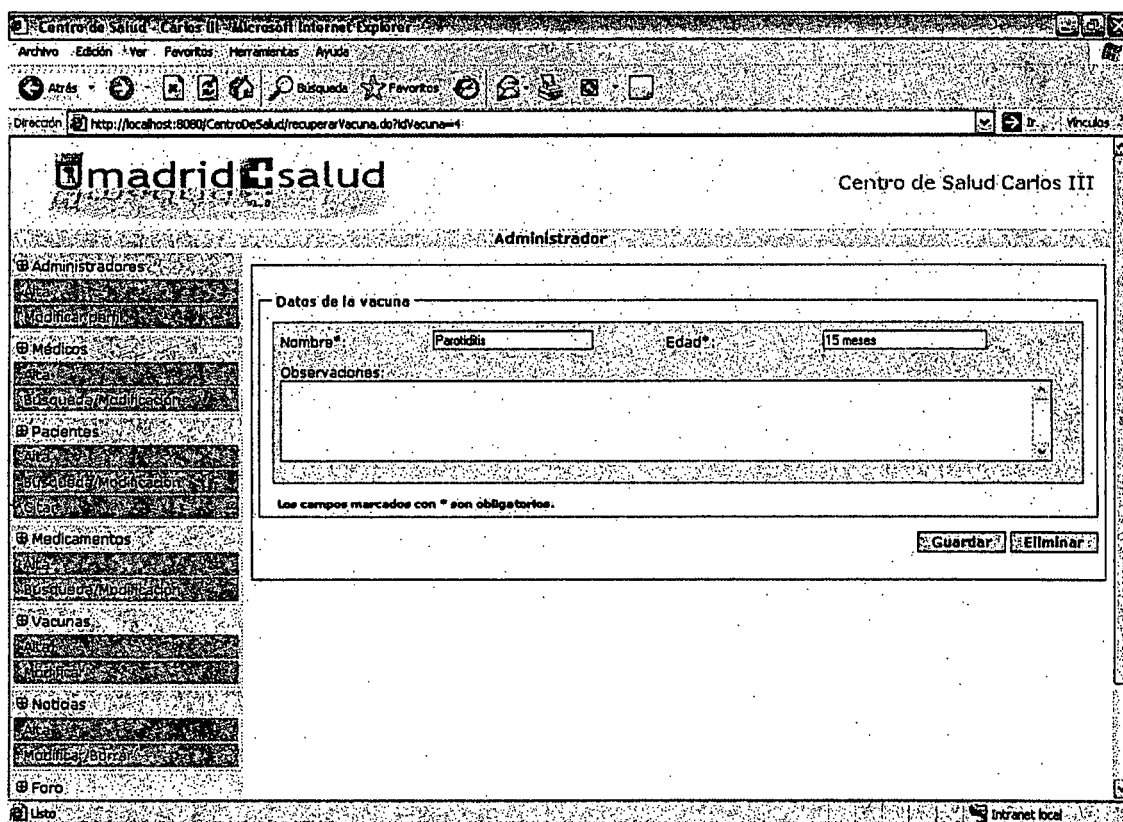


- Figura 263: Resultado accesibilidad Pág. alta administrador prioridad 2 -

Tras comprobar el error de prioridad 2, se observa que se corresponde con la pauta 11, e indica que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador no se podría completar la funcionalidad.

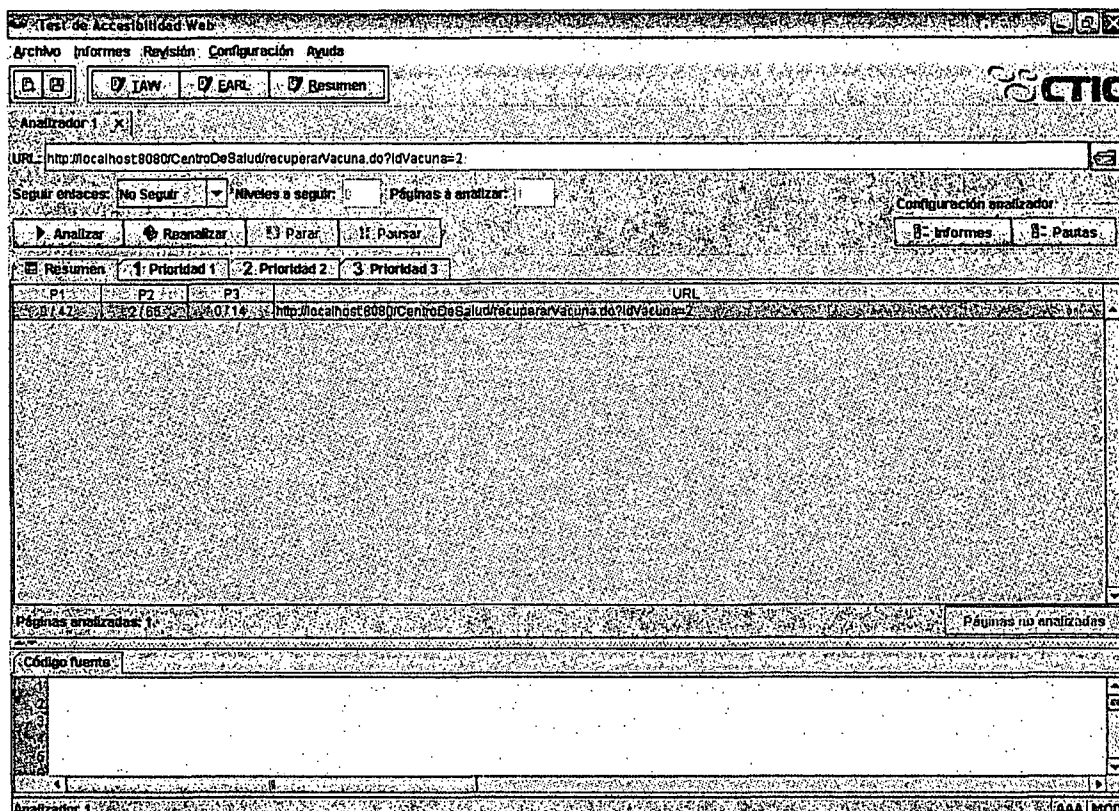
En este caso, el elemento JavaScript se usa para validar que el administrador rellene todos los campos que son obligatorios. Estas validaciones están hechas con el marco Validator de Struts, por lo tanto, en caso de que el administrador desactive JavaScript en su navegador, las validaciones se harán en el servidor. Esto indica que no es aplicable ya que sin JavaScript todo funcionará correctamente, por lo tanto esta página cumple la doble A.

6.3.3.2.- ANÁLISIS DE UNA PÁGINA DE RECUPERAR Y BORRAR



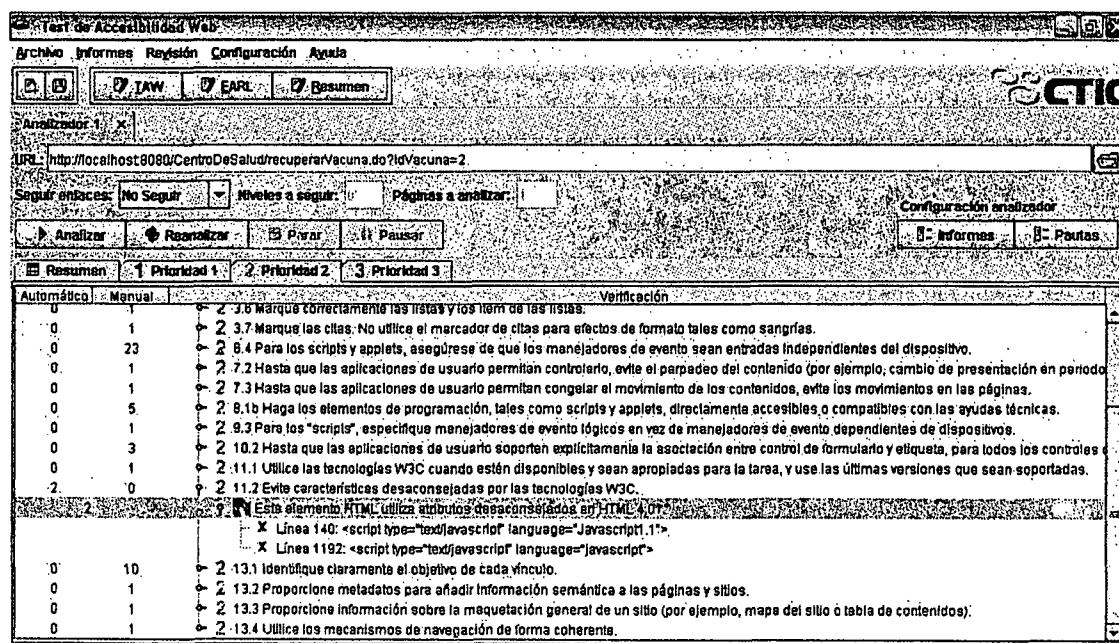
- Figura 264: Recuperar vacuna -

Introducir por ejemplo la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/recuperarVacuna.do?idVacuna=4> y pulsar sobre "Analizar". Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 265: Resultado accesibilidad Pág. recuperar vacuna -

En esta página aparecen dos errores automáticos de prioridad 2, con lo cual, solo cumple A.

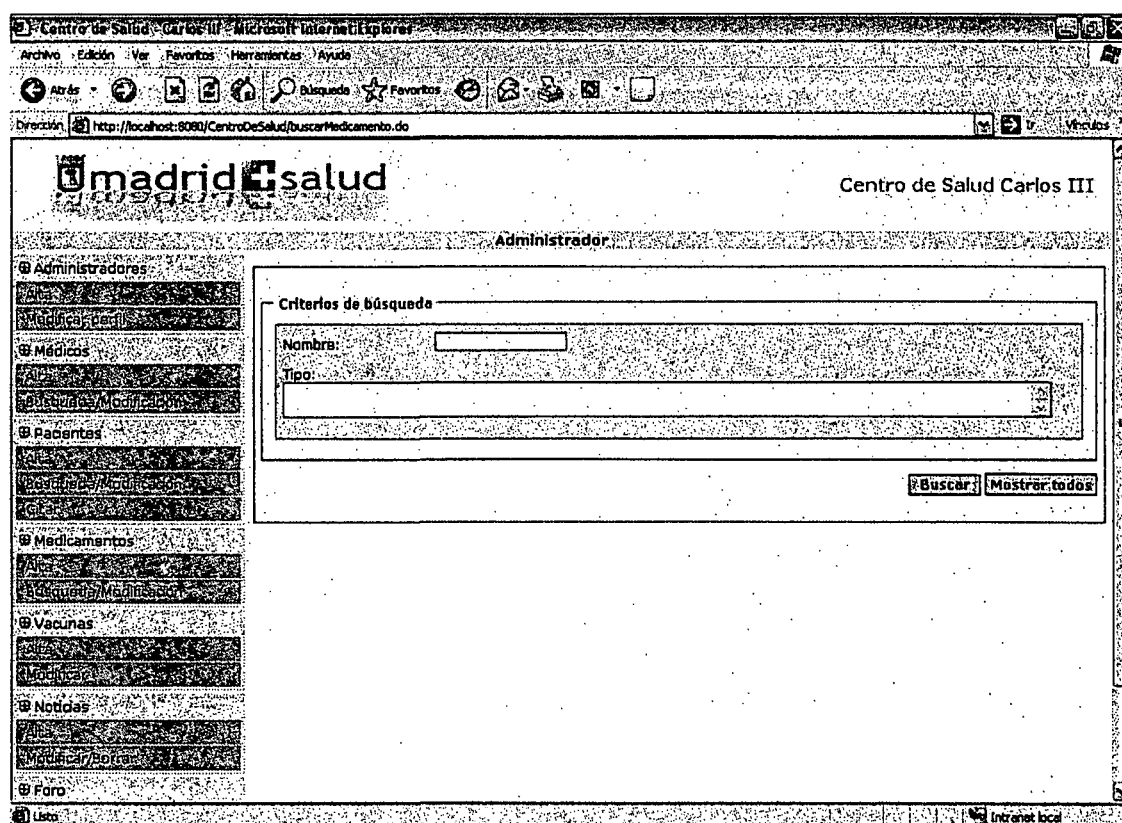


- Figura 266: Resultado accesibilidad Pág. recuperar vacuna prioridad 2 -

Tras comprobar los errores se observa que se corresponden con la pauta 11, e indican que se hace uso de JavaScript, con lo cual en caso de estar desactivado en el navegador no se podría completar la funcionalidad.

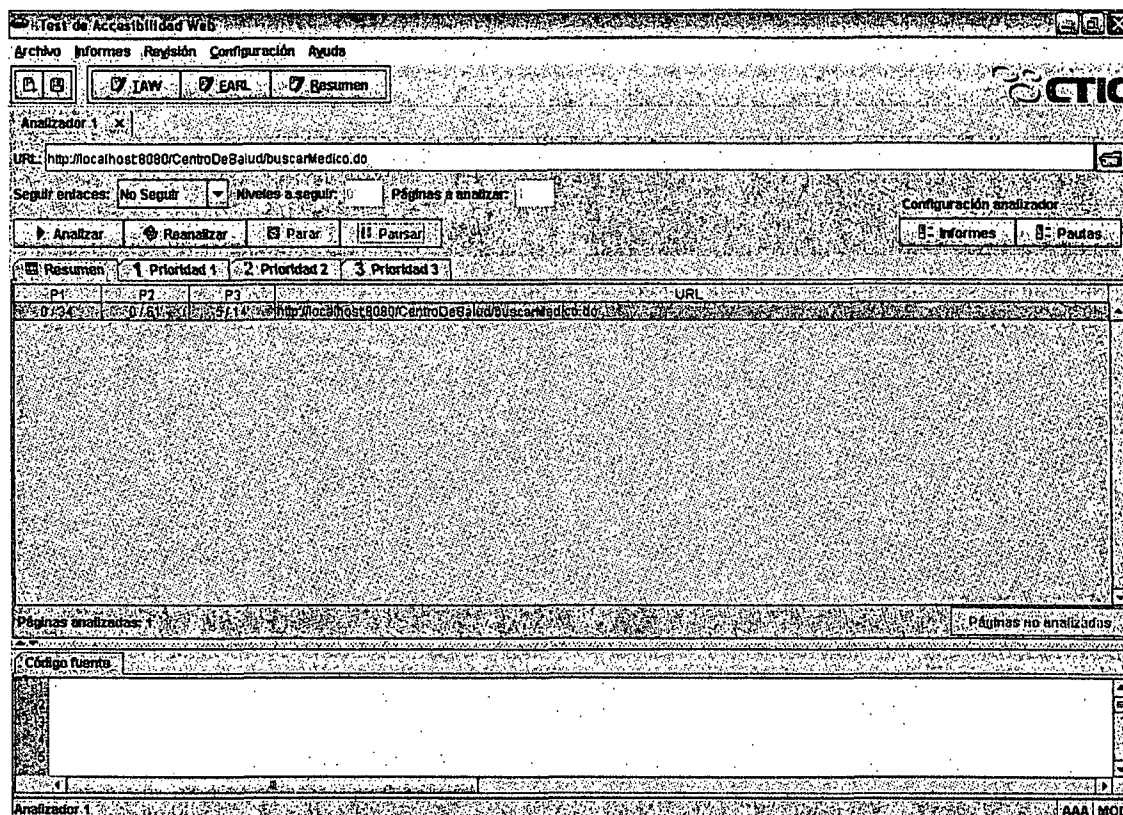
En este caso, el primer elemento JavaScript se usa para validar que el administrador rellene todos los campos que son obligatorios. Estas validaciones están hechas con el marco Validator de Struts, por lo tanto, en caso de que el administrador desactive JavaScript en su navegador, las validaciones se harán en el servidor. Esto indica que no es aplicable ya que sin JavaScript todo funcionará correctamente. El segundo elemento se usa para confirmar la eliminación de una vacuna, por lo tanto no se puede prescindir de él, así que esta página solo cumple A.

6.3.3.3.- ANÁLISIS DE UNA PÁGINA DE BUSQUEDA



- Figura 267: Consultar medicamento -

Introducir por ejemplo la siguiente ruta <http://<host>:<puerto>/CentroDeSalud/buscarMedico.do> y pulsar sobre “Analizar”. Tras realizar el análisis aparecerán los siguientes resultados.



- Figura 268: Resultado accesibilidad Pág. consultar medicamento -

En esta página aparecen cinco errores automáticos de prioridad 3, con lo cual, solo cumple la doble A.



7.- CONCLUSIONES Y LÍNEAS FUTURAS

7.1.- CONCLUSIONES

El proyecto ha cumplido todos los objetivos marcados, además de añadir a medida de su desarrollo nuevas funcionalidades y mejoras que no estaban contempladas, como la modificación y borrado de noticias, páginas con algunos de los elementos de accesibilidad recomendados por la W3C y validaciones a través de JavaScript.

La realización de esta aplicación ha supuesto todo un reto, ya que el uso de nuevas tecnologías que aun no están del todo testadas, ha provocado que surgieran algunas complicaciones de difícil solución, pero que con paciencia y documentación se han ido solucionando.

Los principales problemas han tenido lugar con el uso de la versión 3.2 de Hibernate, ya que algunos tipos de relaciones no funcionan de la manera que dicta el manual, y los administradores de Hibernate no comentan nada acerca de esos problemas, por lo que gracias a los foros que existen en Internet sobre Hibernate, se ha conseguido solucionar todos los problemas de la mejor manera posible.

Finalmente cabe destacar, que al ser un proyecto con tecnologías tan innovadoras y escrito en un lenguaje que apenas se estudia en la Universidad Carlos III de Madrid, ha dado lugar a conocer y aprender más sobre estas herramientas, lo cual es un importante avance para mi vida futura en una empresa.

7.2.- FUTURAS LÍNEAS DE TRABAJO

Aunque es un proyecto bastante completo, siempre cabe lugar a la ampliación y mejora del mismo, así como propuestas de mejora se reflejan las siguientes:

- **Parametrización del seguimiento de Sintrom**, añadiendo una interfaz visual en el menú del administrador, que permita realizar los cambios necesarios en los parámetros que miden la dosis recomendada para un paciente, según diversos factores. De esta forma se conseguiría que el administrador no tuviera que tocar el código de la aplicación directamente, siendo así mucho más fácil ese cambio.
- Realizar las adaptaciones necesarias para cumplir con todo lo que establece la **Ley Oficial de Protección de Datos (LOPD)** en cuanto a datos de carácter personal y sanitario, realizando por consiguiente encriptación de claves, políticas de seguridad a nivel de sistemas y de organización; y firmas de contrato con los responsables de mantenimiento.



- Añadir y contemplar todos los elementos necesarios para **hacer que las páginas sean completamente accesibles** desde el punto de vista de la normativa de W3C.
- **Petición de citas** a través del correo electrónico, facilitando así al paciente este proceso, y logrando que automáticamente se introdujeran dichas citas la agenda del médico.
- **Envío de recetas periódicas** a través del correo electrónico, es decir, que aquellos pacientes que necesiten solicitar recetas de un medicamento cada tres meses, no tengan que acudir al Centro de Salud para recogerlas, sino que automáticamente se reenvíen a su correo electrónico y de esta manera puedan acceder a ellas.



8.- BIBLIOGRAFÍA

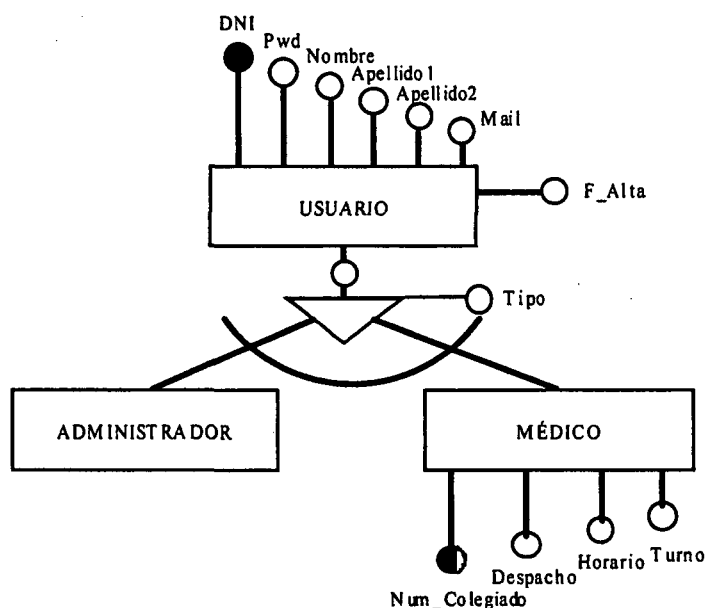
- Adoración de Miguel Castaño, Paloma Martínez Fernández, Elena Castro Galán, José María Caverio Barca, Dolores Cuadra Fernández, Ana María Iglesias Maqueda, Carlos Nieto Lázaro (2001), **Diseño de Bases de Datos**, Editorial Ra-Ma.
- Bruce Eckel (2003), **Thinking Java**, Prentice Hall.
- Javier García de Jalón, José Ignacio Rodríguez, Iñigo Mingo, Aitor Imaz, Alfonso Brazales, Alberto Larzabal, Jesús Calleja, Jon García (2000), **Aprenda Java como si Estuviera en Primero**, Universidad de Navarra.
- Chuck Cavaness, **Jakarta Struts**, Editorial Anaya.
- Hibernate, **Hibernate Reference Documentation Version 3.2.1**.
- Christian Bauer, Gavin King (2005), **Hibernate In Action**, Hanning.
- Patrick Peak, Nick Heudecker (2006), **Hibernate Quickly**, Hanning.
- Francesc Rosés Albiol (2004), **Introducción a Hibernate**.
- Sebastián Hennebrüder (2005), **Primeros Pasos Usando Struts e Hibernate**.
- Ignacio Díaz Asenjo, **Hibernate: Persistencia de Objetos Java**, Doctorado Tecnología de las Comunicaciones, Universidad Carlos III de Madrid.
- Erick A. Meyer (2000), **Cascading Style Sheets, The Definitive Guide**, O'Reilly
- David Flannagan (1997), **JavaScript, The Definitive Guide**, O'Reilly.
- Cliff Wootton (2001), **JavaScript Programmer's Reference**, Wrox.
- IEEE 1074 (1997), **Standard for Developing Software Life Cycle Processes Document**.
- Página de referencia de Struts, [http:// struts.apache.org](http://struts.apache.org)
- Página de referencia de Hibernate, <http://www.hibernate.org>
- Página de referencia de MySQL, <http://dev.mysql.com/>



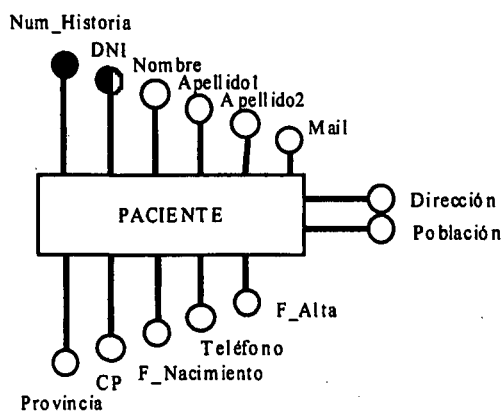
- Página con diversos tutoriales sobre Java, Bases de Datos..., <http://www.programacion.com>
- Enciclopedia Virtual, <http://es.wikipedia.org/wiki/Portada>
- Página con tutoriales sobre Hibernate, Struts, Tomcat, Eclipse..., <http://laliluna.de>
- Página con diversos tutoriales sobre Java, <http://www.javahispano.org>
- Revista Teóricos, contiene un artículo sobre motores de persistencia, <http://www.ufg.edu.sv/ufg/theorethikos/Julio04/sumario.html>
- Jabato, apuntes de contenido técnico, <http://www.telefonica.net/web2/robermh/index.html>
- VnsJava, blog que contiene un artículo sobre “*Como hacer un JDBC Datasource con Database Connection Pool en Tomcat e Hibernate*”, http://vnsjava.blogspot.com/2005_10_16_archive.html
- Normativa de accesibilidad WAI en castellano, http://www.dicapnet.es/web_accesible/wcag10/WAI-WEBCONTENT-19990505_es.html
- Asociación española para el diseño de accesibilidad, <http://www.sidar.org>
- Verificador de hojas de estilo en cascada (CSS) y documentos (X)HTML con hojas de estilo, <http://jigsaw.w3.org/css-validator/>

ANEXO

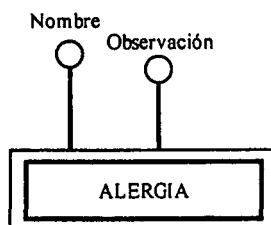
En el siguiente anexo se adjuntan las clases junto con sus atributos pertenecientes al diagrama E/R, en cual no se han añadido para una mayor claridad.



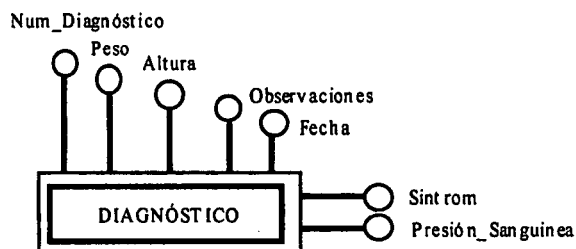
- Figura 269: Jerarquía usuario -



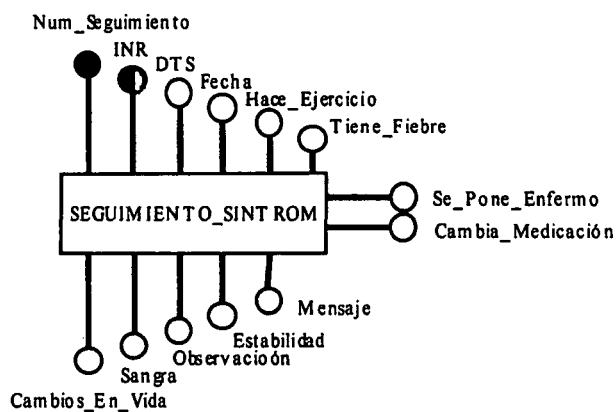
- Figura 270: Clase paciente -



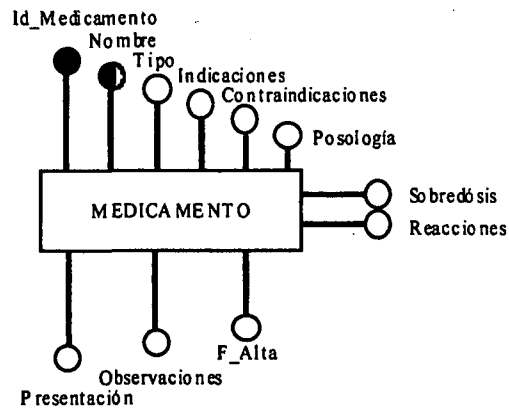
- Figura 271: Clase alergia -



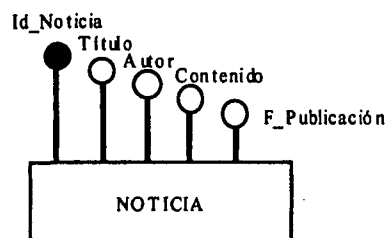
- Figura 272: Clase diagnóstico -



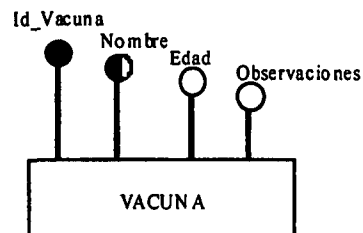
- Figura 273: Clase seguimiento de Sintrom -



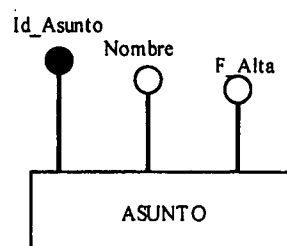
- Figura 274: Clase medicamento -



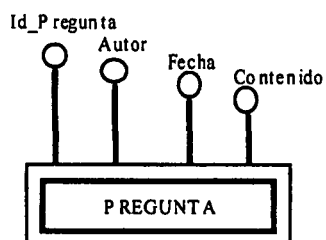
- Figura 275: Clase noticia -



- Figura 276: Clase vacuna -



- Figura 277: Clase asunto -



- Figura 278: Clase pregunta -